CS2LAP: Logic and Prolog 2000/2001
Sample Solutions for Coursework 5 (Questions 1–5)

Question 1.

Goal: $\underline{A \Rightarrow \neg\neg A}$

(1)	$A \Rightarrow \neg\neg A$	from subcomputation

> Goal: $\underline{\neg\neg A}$
>
> | | | |
> |---|---|---|
> | (1.1) | A | assumption |
> | (1.2) | $\neg A \Rightarrow A$ | from subcomputation |
>
> > Goal: $\underline{A}$
> >
> > | | | |
> > |---|---|---|
> > | (1.2.1) | $\neg A$ | assumption |
> > | (1.2.2) | A | from (1.1) |
>
> | | | |
> |---|---|---|
> | (1.3) | $\neg A \Rightarrow \neg A$ | from subcomputation |
>
> > Goal: $\underline{\neg A}$
> >
> > | | | |
> > |---|---|---|
> > | (1.3.1) | $\neg A$ | assumption |
> > | (1.3.2) | $\neg A$ | from (1.3.1) |
>
> | | | |
> |---|---|---|
> | (1.4) | $\neg\neg A$ | ($\neg$I) with (1.2) and (1.3) |

Question 2. A model with the domain of the real numbers where the function h is interpreted as "multiplication with -1" and the predicate A is interpreted as the "strictly greater"-relation $<$ satisfies the formula $\forall x.\exists y.(A(x,y) \wedge \neg A(h(x), h(y)))$.

$$\begin{aligned} \mathcal{D} &= \mathbb{R} \\ \mathcal{I}(h) &= \{x \mapsto -1 \cdot x\} \\ \mathcal{I}(A) &= \{(x,y) \in \mathbb{R}^2 \mid x < y\} \end{aligned}$$

For every real number x there exists another real number y that is strictly greater than x. For the same x and y the number $-x$ is not strictly greater than $-y$. Hence, the above sentence is satisfied by the model $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$.

Question 3.

(a)

$\vdash (\forall x.\neg P(x)) \Rightarrow (\neg\forall x.P(x))$

rule for $\Rightarrow$

$\forall x.\neg P(x) \vdash \neg\forall x.P(x)$

rewrite goal

$\forall x.\neg P(x) \vdash \exists x.\neg P(x)$

rewrite with free variables

$\forall x.\neg P(x),\ \neg P(u) \vdash \neg P(v)$

rewrite

continues next column

$\forall x.\neg P(x),\ P(u) \Rightarrow \bot \vdash P(v) \Rightarrow \bot$

rule for $\Rightarrow$

$\forall x.\neg P(x),\ P(u) \Rightarrow \bot,\ P(v) \vdash \bot$

rule for atoms

$\forall x.\neg P(x),\ P(v) \vdash P(u)$

substitution $\{u = v\}$

$\forall x.\neg P(x),\ P(v) \vdash P(v)$

success

(b) A countermodel for a formula is a model in which that formula isn't satisfied. There are many countermodels for $(\neg\forall x.P(x)) \Rightarrow (\forall x.\neg P(x))$. We give two very different examples here.

- First solution. We choose a model with the natural numbers as the domain and interpret the predicate P as "is a prime number".

$$\begin{aligned} \mathcal{D} &= \mathbb{N} \\ \mathcal{I}(P) &= \{n \in \mathbb{N} \mid n \text{ is prime}\} \end{aligned}$$

  Clearly, in this model the above formula is not true, because even though not all natural numbers are prime it is not the case that all natural numbers are not prime.

- Second solution. Consider as a domain the set of all apples and pears (both sets can't be empty) and interpret P as "is a pear".

$$\begin{aligned} \mathcal{D} &= \{x \mid x \text{ is an apple or a pear}\} \\ \mathcal{I}(P) &= \{x \mid x \text{ is a pear}\} \end{aligned}$$

  Also in this model the formula in question doesn't hold, because even though not everything is a pear it is not the case that everything is not a pear.

Question 4.

$$\{\, \forall x.(\mathit{Man}(x) \Rightarrow \mathit{Mortal}(x)),\ \mathit{Man}(\mathit{Socrates}),\ \mathit{Mortal}(\mathit{Socrates}) \Rightarrow \perp \,\}$$

Question 5.

(a)
```
?- append( [elephant | X], X, [elephant]).
X = []
Yes
```
If we want to append two lists to get the list `[elephant]` this surely isn't possible if either of them has more than one element. Hence, in the example above `X` can only be `[]`.

(b)
```
?- Y + 17 = 3 * 4 + 17, \+ (Y > 15).
Y = 3*4
Yes
```
For the first subgoal we have to match `Y + 17` with `3 * 4 + 17`. Hence, `Y` is instantiated with the term `3 * 4`. Note that `Y` is *not* `12`, because there's no arithmetic evaluation (`is`-operator) involved. After `Y` has been matched with `3 * 4` the arithmetic relation `Y > 15` fails (because 12 isn't greater than 15). Hence, the second subgoal also succeeds.

(c)
```
?- Z * 17 = 3 + 4 * 17.
No
```
The principal operator of the term on the right-hand side is `+`. Therefore, it cannot match the term on the left-hand side whose principal operator is `*`.

Sample solutions for **Questions 6** and **7** are available as Prolog programs from the CS2LAP course web site:

http://www.dcs.kcl.ac.uk/teaching/units/cs2lap/