Informatics and Mathematical Modelling
DTU

02280 Data Logic spring 2004

Solutions to most exercises in *Data Logic* chapters 1 - 3

Exercise 1.1 Bike lights electronic circuit specification

$(p_1 \wedge p_4 \wedge p_5) \leftrightarrow p_6$
$(p_1 \wedge p_4 \wedge p_7) \leftrightarrow p_8$
$p_3 \leftrightarrow \neg p_4$
$p_1 \leftrightarrow \neg p_2$

Exercise 1.3 Bike lights specification using implication and negation, only

$p_1 \rightarrow (p_4 \rightarrow (p_5 \rightarrow p_6))$
$p_6 \rightarrow p_1$
$p_6 \rightarrow p_4$
$p_6 \rightarrow p_5$

$p_1 \rightarrow (p_4 \rightarrow (p_7 \rightarrow p_8))$
$p_8 \rightarrow p_1$
$p_8 \rightarrow p_4$
$p_8 \rightarrow p_7$

$p_3 \rightarrow \neg p_4$
$\neg p_4 \rightarrow p_3$

$p_1 \rightarrow \neg p_2$
$\neg p_2 \rightarrow p_1$

Exercise 2.1 Compositionality principle

Imperative programming languages challenge the compositionality principle.

Exercise 2.2 Calculation of truth value

With the stated interpretation one gets

$[\![p \rightarrow (q \rightarrow p)]\!] = [\![\rightarrow]\!]([\![p]\!], [\![(q \rightarrow p)]\!]) = [\![\rightarrow]\!]([\![p]\!], [\![\rightarrow]\!]([\![q]\!], [\![p]\!])) =$
$[\![\rightarrow]\!](F, [\![\rightarrow]\!](T, F)) = [\![\rightarrow]\!](F, F) = \mathrm{T}$

Exercise 2.3 Number of interpretations

2^n, where n is the number of proper atomic propositions.

Exercise 2.4

a) There are four 1-argument Boolean functions: (i) The constant function giving F, (ii) the constant function giving T, (iii) the identity function, (iiii) negation. Only negation has an operator name.

b) 2^{2^n}. (Hint: Observe that there are 2^n different rows in the possible truth tables in case of n arguments.)

Exercise 2.5

p is satisfiable. $p \wedge (\neg p)$ is unsatisfiable.

Exercise 2.7

Satisfiable and unsatisfiable sentences are disjoint (i.e. non-overlapping) classes. Valid sentences are included in the class of satisfiable sentences.

Exercise 2.8

$s \models \textit{false}$ iff s *is unsatisfiable* iff $\neg s$ *is valid*

Exercise 2.9

$\models$ is a metalanguage operator, whereas $\rightarrow$ is an operator in propositional logic.

$s_1 \models s_2$ iff $\models (s_1 \rightarrow s_2)$

Exercise 2.10

Analogous to 2.9.

Exercise 2.11 Validating logical equivalences

The logical equivalences can be proved by considering the 8 different interpretations.

Exercise 2.12 Contrapositioning of implication sentences

The two sentences are logically equivalent and therefore contains the same logical meaning content according to classical logic.

Exercise 2.13 (extra) If-then-else by means of binary logical operators

1) $(\varphi \wedge \psi_1) \vee (\neg\varphi \wedge \psi_2)$ (other solutions are possible)

2) $\neg(\neg(\varphi \wedge \psi_1) \wedge (\neg(\neg\varphi \wedge \psi_2)))$

3) No, implication cannot express negation.

Exercise 2.14 (extra) Binary logical operators by means of if-then-else

1) if p then false else true

2) if p then q else false

3) if p then true else q

4) if p then q else true

Exercise 3.1 Soundness of logical system

Soundness of Axiomatic logic in the stated form is established by verifying that the axiom schemes produce valid sentences. This is done by truth table inspection (check of possible interpretations). Moreover it is to be verified straightforwardly by truth table inspection that the conclusion in the modus ponens rule is true in any interpretation where the two premises are true.

Exercise 3.2 Sample formal proof in axiomatic logic

From axioms schemes one can obtain instances

$p \to (p \to p)$

$p \to ((p \to p) \to p)$

$p \to ((p \to p) \to p) \to (p \to (p \to p)) \to (p \to p)$

These instances are then used with modus ponens:

$$\frac{p \to ((p \to p) \to p) \to (p \to (p \to p)) \to (p \to p) \qquad p \to ((p \to p) \to p)}{(p \to (p \to p)) \to (p \to p)}$$

and further

$$\frac{(p \to (p \to p)) \to (p \to p) \qquad (p \to (p \to p))}{p \to p}$$

This is the shortest formal proof of $p \to p$ in the considered logical system.

Exercise 3.3 Implication and negation suffice

Use the equivalences:

$(P \wedge Q) \quad \equiv \quad \neg(\neg P \vee \neg Q) \quad \equiv \quad \neg(P \to \neg Q)$

$(P \vee Q) \quad \equiv \quad \neg(\neg P) \vee \neg Q \quad \equiv \quad (\neg P) \to Q$

Exercise 3.4 Sample formal proof in axiomatic logic

To the sentences from exercise 1.3 constituting the logical systems specification we may add the observation sentences p_6 and $\neg p_8$.

Then it is easy to set up a proof of $(p_7 \to p_8)$ using modus ponens 3 times.

Now using the lemma $((P \to Q) \to (\neg Q \to \neg P))$ with modus ponens one gets $(\neg p_8 \to \neg p_7)$, from which $\neg p_7$ obtains by modus ponens.

Moral: Formal proofs in axiomatic logic tend to be long and tedious unless adequate lemmas (proved separately) are made available. For practical purposes better logical systems are available.