Informatics and Mathematical Modelling
DTU

# 02280 Data-Logic  Spring term 2003

## Solutions to most exercises in *Data Logic* chapters 4 - 6

**Exercise 4.2** Classification of clauses using propositional logic

(a)

$clause \leftrightarrow horn \vee indef$
$horn \rightarrow \neg indef$
$horn \leftrightarrow goal \vee definite$
$goal \rightarrow \neg definite$

$goal \leftrightarrow empty \vee neg$
$empty \rightarrow \neg neg$

$pos \rightarrow definite \vee indef$

$unit \rightarrow horn$

Observe that the propositional logic when used for classification fails to make a distinction between individual and class (cf. the singular empty clause).

**Exercise 4.4** Discarding of non-definite clauses

It is not possible in general to turn non-definite clauses into definite ones by introducing a new predicate *nonp* for $\neg p$, since the definition

$$nonp \leftrightarrow \neg p$$

in itself gives rise to non-definite clauses.

**Exercise 4.5** Resolution incomplete for direct proofs

The valid sentence $p \vee (\neg p)$ cannot be produced from the resolution rule. Therefore, instead, from the denial $p \wedge (\neg p)$, giving the two clauses $p$ and $(\neg p)$, the empty clause is to be produced directly by resolution.

**Exercise 5.3**

For instance the rule

$$\frac{Every\ P\ is\ Q \quad Every\ Q\ is\ R}{Every\ P\ is\ R}$$

is sound since the conclusion is true in all interpretations in which both of the premises are true:

If $[\![P]\!] \subseteq [\![Q]\!]$ and $[\![Q]\!] \subseteq [\![R]\!]$ then $[\![P]\!] \subseteq [\![R]\!]$.

Similarly can be argued for the other rules.

Missing rules:

$$\frac{Some\ P\ is\ Q}{Some\ Q\ is\ P}$$

and

$$\frac{X\ is\ P \quad X\ is\ Q}{Some\ P\ is\ Q}$$

and

$$\frac{}{Every\ P\ is\ P}$$

**Exercise 5.4** Counterintuitive non-logical consequence

Consider the special case that $p$ and $q$ are empty classes.

**Exercise 5.5** A formal proof in class logic

A formal proof (though not the shortest one)

$$\frac{No\ mammal\ is\ fish}{No\ fish\ is\ mammal}$$

$$\frac{Every\ shark\ is\ fish \quad No\ fish\ is\ mammal}{No\ shark\ is\ mammal}$$

$$\frac{No\ shark\ is\ mammal}{No\ mammal\ is\ shark}$$

$$\frac{Every\ whale\ is\ mammal \quad No\ mammal\ is\ shark}{No\ whale\ is\ shark}$$

$$\frac{No\ whale\ is\ shark}{No\ shark\ is\ whale}$$

**Exercise 5.7**

As an example in the considered class logic it is not possible to express "every P is Q or R" (e.g. every person is male or female).

**Exercise 6.1** Example in classification logic

$dog(nelly)$

becomes "Nelly is dog".

$cat(freddy)$

becomes “Freddy is cat”.

$pet(X) \leftarrow cat(X)$

becomes “Every cat is pet”.

$pet(X) \leftarrow dog(X)$

becomes “dog is pet”.

$likes(nelly, X) \leftarrow dog(X)$

Cannot be properly handled in class logic since it comprises a binary predicate.

**Exercise 6.2** Examples expressed in DATALOG

*All cats and dogs like each other.*

becomes

$like(X, Y) \leftarrow cat(X) \wedge dog(Y)$
$like(Y, X) \leftarrow cat(X) \wedge dog(Y)$

*All cats and dogs like milk.*

$like(X, milk) \leftarrow cat(X)$
$like(X, milk) \leftarrow dog(X)$

*No cats are dogs.*

strictly cannot be handled in DATALOG. However sometimes is used a formulation like

$error(X) \leftarrow cat(X) \wedge dog(X)$

intended to capture erroneous samples.

**Exercise 6.3** Sample atomic deductive closure

The atomic deductive closure is

$dog(nelly)$
$cat(freddy)$
$pet(nelly)$
$pet(freddy)$
$likes(nelly, nelly)$

**Exercise 6.4** Transitive closure

The relation *ancestor* is the transitive closure of the relation *parentof*.

**Exercise 6.5** Database operations in DATALOG

Assume that the contents of two database relations $r_1$ and $r_2$ in the logical context is specified as atomic clauses:

$\mathrm{r}_1(c_{11}, ..., c_{1n})$
$\mathrm{r}_1(c_{21}, ..., c_{2n})$ ... etc.

Union:

$$r(X_1, ..., X_n) \leftarrow r_1(X_1, ..., X_n)$$
$$r(X_1, ..., X_n) \leftarrow r_2(X_1, ..., X_n)$$

Intersection:

$$r(X_1, ..., X_n) \leftarrow r_1(X_1, ..., X_n) \wedge r_2(X_1, ..., X_n)$$

It is not possible to express difference in the present version of DATALOG. However, later is introduced the so called “negation as failure” rule, which makes it possible to handle difference.

**Exercise 6.6**

In this exercise one is to verify that the set of atomic ground atoms forming the deductive closure (cf. exercise 6.3) are the (only) ground atomic logical consequences. This is done by examination of the Herbrand interpretations.

**Extra Exercise 1** Predicate logical specification of won/lost in chess and similar games

The predicate

$$move(S1, T, S2)$$

expresses that $T$ is a legal move in the chessboard situation $S1$, leading to the situation $S2$. The predicate

$$check(S)$$

expresses that the situation is ‘check’.

The two predicates can be defined by very complicated specifications (logic programs) formalising all of the rules for the chess pieces. But here you just consider the predicates as given and use them when specifying the following chess concepts as logic predicates:

1. The situation is checkmate.
2. The situation is (a) won, (b) lost, (c) a draw (especially perpetual check) for the player in move.
3. The situation is mate in 2 moves.

Is it decidable whether the initial situation is wonC for white (or black), or a draw? How does this question relate to solutions to 2 ?

Observe that these predicates apply to a whole class of 2-person games including tic-tac-toe as a simple case.