Informatics and Mathematical Modelling
DTU

02280 Data Logic spring 2003

Solutions to most exercises in *Data Logic* chapters 7 - 10

Exercise 7.2 BNF-grammar for predicate logic

A simple grammar, which does not account for operator priorities and associativity properties of operators, may look as follows:

$\langle formula \rangle ::= \langle compformula \rangle \mid \langle quantformula \rangle \mid \langle atomformula \rangle$

$\langle compformula \rangle ::=$
$\quad (\langle formula \rangle \wedge \langle formula \rangle) \mid$
$\quad (\langle formula \rangle \vee \langle formula \rangle) \mid$
$\quad \neg \langle formula \rangle$

$\langle quantformula \rangle ::= \forall \langle variable \rangle (\langle formula \rangle)$

$\langle atomformula \rangle ::= \langle pred \rangle (\langle termlist \rangle)$

$\langle termlist \rangle ::= \langle term \rangle \mid \langle term \rangle , \langle termlist \rangle$

$\langle term \rangle ::= \langle const \rangle \mid \langle variable \rangle \mid \langle functor \rangle (\langle termlist \rangle)$

$\langle pred \rangle ::= \langle identifier \rangle$

$\langle const \rangle ::= \langle identifier \rangle$

$\langle functor \rangle ::= \langle identifier \rangle$

$\langle variable \rangle ::= \langle identifier \rangle$

Exercise 8.3 Equivalence laws for quantifiers

$$\forall X(p(X) \wedge q(X)) \equiv (\forall X p(X)) \wedge (\forall X q(X))$$

This equivalence is proved by considering the definition of truth value assignment for quantifiers.

$$\forall X(p(X) \vee q(X)) \not\equiv (\forall X p(X)) \vee (\forall X q(X))$$

This non-equivalence is verified by considering an interpretation in which p and q is not true for all elements in the domain, but where either p or q is true.

The laws for existential quantification can be studied by using the definitional equivalence $\exists X p(X) \equiv \neg \forall X \neg p(X)$.

Thereby one gets the complementary results

$$\exists X(p(X) \wedge q(X)) \not\equiv (\exists X p(X)) \wedge (\exists X q(X))$$

and

$$\exists X(p(X) \vee q(X)) \equiv (\exists X p(X)) \vee (\exists X q(X))$$

Exercise 8.4 Sample rewriting to definite clauses

$$\forall X \forall Y (a(X,Y) \leftarrow (\exists Z (a(X,Z) \wedge a(Z,Y)) \vee p(X,Y)))$$

is equivalent with

$$\forall X \forall Y (a(X,Y) \vee (\neg(\exists Z (a(X,Z) \wedge a(Z,Y)) \vee p(X,Y))))$$

is equivalent with

$$\forall X \forall Y (a(X,Y) \vee (\forall Z (\neg a(X,Z) \vee \neg a(Z,Y)) \wedge \neg p(X,Y))))$$

is equivalent with

$$\forall\ a(X,Y) \vee ((\neg a(X,Z) \vee \neg a(Z,Y)) \wedge \neg p(X,Y)))$$

is equivalent with

$$\forall\ a(X,Y) \vee \neg p(X,Y)$$
$$\forall\ a(X,Y) \vee (\neg a(X,Z) \vee \neg a(Z,Y))$$

is equivalent with

$$a(X,Y) \leftarrow p(X,Y)$$
$$a(X,Y) \leftarrow a(X,Z) \wedge a(Z,Y)$$

Thus one obtains two definite clauses.

Exercise 8.5 Sample rewriting to clauses

$$\forall (subset(X,Y) \quad \rightarrow \quad (\forall Z (in(Z,X) \rightarrow in(Z,Y)))$$

can be rewritten into one definite clause.

$$\forall (subset(X,Y) \quad \leftarrow \quad (\forall Z (in(Z,X) \rightarrow in(Z,Y)))$$

Rewriting to clause form calls for introduction of a two-argument Skolem function. The resulting clauses comprise one definite and one indefinite clause.

Exercise 10.1 Laws for term substitutions

Associativity:

$$(\theta_1 \circ \theta_2) \circ \theta_3 \ = \ \theta_1 \circ (\theta_2 \circ \theta_3)$$

follows by the rewritings

$$t((\theta_1 \circ \theta_2) \circ \theta_3) \ = \ (t(\theta_1 \circ \theta_2)\theta_3) \ = \ ((t(\theta_1)\theta_2)\theta_3) \ = \ ((t\theta_1)(\theta_2 \circ \theta_3) \ = \ t(\theta_1 \circ (\theta_2 \circ \theta_3))$$

Similarly is proved: $\theta \circ \epsilon \ = \ \theta \ = \ \epsilon \circ \theta$.

Exercise 10.5

Unification of the pair of terms $p(X,X)$ and $p(Y,f(Y))$ fails on 'occur-check' since the term equation $Y = f(Y)$ has no solution.

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.

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

A Solution

Let us make life easier by introducing the auxiliary predicate m:

$$\forall\ m(S_1, S_2) \leftrightarrow \exists T move(S_1, T, S_2)$$

A position is won for a player if there exists a move for him leading to a lost position for the other player.

A position is lost for a player if he is mate or if all possible moves for him lead to a position being won for the other player.

This is formalised as follows

$$\forall\ checkmate(S) \leftrightarrow check(S) \wedge \neg\exists S' m(S, S')$$

$$\forall\ won(S) \leftrightarrow \exists S'(m(S, S') \wedge lost(S'))$$

$$\forall\ lost(S) \leftrightarrow (checkmate(S) \vee \exists S'(m(S, S') \wedge \forall S'(m(S, S') \rightarrow won(S'))))$$

$$\forall\ draw(S) \leftrightarrow \neg won(S) \wedge \neg lost(S)$$

$$\forall\ matein2(S) \leftrightarrow$$
$$\exists S_1(m(S, S_1) \wedge \forall S_2(m(S_1, S_2) \rightarrow \exists S_3(m(S_2, S_3) \wedge checkmate(S_3))))$$