Informatics and Mathematical Modelling
DTU

02280 Data-Logic Spring term 2002

Solutions to most exercises in *Data Logic* chapters 11 - 13

Exercise 11.3 Logic program for factorial

$fac(o, s(o)).$
$fac(s(X), Z) \leftarrow fac(X, Y), mult(s(X), Y, Z).$

Exercise 11.4 Logic program for addition of integers

$plus(o, Y, Y).$
$plus(s(X), s(Y), s(Z)) \leftarrow plus(X, s(Y), Z).$
$plus(s(X), o, s(X)).$
$plus(s(X), p(Y), Z) \leftarrow plus(X, Y, Z).$
$plus(p(X), s(Y), Z) \leftarrow plus(X, Y, Z).$
$plus(p(X), o, p(X)).$
$plus(p(X), p(Y), p(Z)) \leftarrow plus(X, p(Y), Z).$

Exercise 12.7 Logic program for sublist of a list

A solution to this problem may be achieved with the versatile *append*

$sublist(L1, L2) \leftarrow append(Z, Y, L2), append(X, L1, Z).$

Observe the ordering of clauses in the body ensuring that applications of *append* terminates.

Exercise 12.8 Logic program for reversing a list

Observe how the induction principle for construction of list logic programs is often applied in the below problems.

$reverse([], []).$
$reverse([X|T], L) \leftarrow reverse(T, Tr), append(Tr, [X], L).$

This solution is called naive reverse. A more efficient solution is obtained with use of difference lists.

Exercise 12.9 Logic program palindrome

A palindrome is a word etc. that reads the same backwards and forwards, e.g. 'anna' and 'sos'.

Check that a list forms a palindrome can be done simply by

$pal(L) \leftarrow reverse(L, L).$

or with

$pal([]).$
$pal([X]).$
$pal([X|T]) \leftarrow append(L, [X], T), pal(L).$

Generation of an infinite stream of palindromes in the form of non-ground lists can be achieved by the reformulation (swapping in body)

$pal([]).$
$pal([X]).$
$pal([X|T]) \leftarrow pal(L), append(L, [X], T).$

Exercise 12.10 Logic program for permutation of a list

The relationship $perm(l1, l2)$ is to express that the list $l2$ is formed from $l1$ by swapping of elements.

$perm([], []).$
$perm([X|T], L) \leftarrow perm(T, T1), insert(X, T1, L).$

with the auxiliary

$insert(X, [], [X]).$
$insert(X, [Y|T], [X, Y|T]).$
$insert(X, [Y|T], [Y|T1]) \leftarrow insert(X, T, T1).$

Exercise 13.5 Logic program for Fibonacci number series

The series: 1, 1, 2, 3, 5, 8, ... (starting in position 0) may be formed by

$fib(o, s(o)).$
$fib(s(o), s(o)).$
$fib(s(s(X)), Z) \leftarrow fib(s(X), Y1), fib(X, Y2), plus(Y1, Y2, Z).$

It is also possible to formulate a more efficient iterative program version in definite clauses.

Exercise 13.6 Quicksort

$qsort([], []).$
$qsort([X|T], L) \leftarrow$
$\quad split(T, X, L1, L2),$
$\quad qsort(L1, L1s), qsort(L2, L2s),$
$\quad append(L1s, [X|L2s], L).$

$split([], X, [], []).$
$split([Y|T], X, [Y|L1], L2) \leftarrow less(Y, X), split(T, X, L1, L2).$
$split([Y|T], X, L1, [Y|L2]) \leftarrow lesseq(X, Y), split(T, X, L1, L2).$