FIT2014 - Assignment 1

James Austin

August 19, 2014

1 Problem 0

1.1 Propositional Logic

$\neg((P_{VA} \wedge P_{WA}) \vee (P_{VB} \wedge P_{WB}) \vee (P_{VC} \wedge P_{WC}) \vee (P_{XA} \wedge P_{YA}) \vee (P_{XB} \wedge P_{WB}) \vee (P_{XC} \wedge P_{YC}) \vee (P_{WA} \wedge P_{YA}) \vee (P_{WB} \wedge P_{WB}) \vee (P_{WC} \wedge P_{YC}) \vee (P_{YA} \wedge P_{ZA}) \vee (P_{YB} \wedge P_{WB}) \vee (P_{YC} \wedge P_{ZC}))$

1.2 Predicate Logic

For every variable there exists a register assigned to it.
$\forall variable(Q) \exists (allocateVariable(Q, R) \wedge register(R))$

For every conflict, there is no allocation of variables Q1 and Q2 to the same register.
$\forall conflict(Q1, Q2) \exists \neg (allocateVariable(Q1, R) \wedge allocateVariable(Q2, R))$

2 Problem 2

There exists 27 solutions. There are.

V = a, W = b, X = a, Y = c, Z = a
V = a, W = b, X = a, Y = c, Z = a
V = a, W = b, X = a, Y = c, Z = a
V = a, W = b, X = a, Y = c, Z = a
V = a, W = b, X = a, Y = c, Z = b
V = a, W = b, X = c, Y = a, Z = b
V = a, W = b, X = c, Y = a, Z = c
V = a, W = c, X = a, Y = b, Z = a
V = a, W = c, X = a, Y = b, Z = c
V = a, W = c, X = b, Y = a, Z = b
V = a, W = c, X = b, Y = a, Z = c
V = b, W = a, X = b, Y = c, Z = a
V = b, W = a, X = b, Y = c, Z = b
V = b, W = a, X = c, Y = b, Z = a

V = b, W = a, X = c, Y = b, Z = c
V = b, W = c, X = a, Y = b, Z = a
V = b, W = c, X = a, Y = b, Z = c
V = b, W = c, X = b, Y = a, Z = b
V = b, W = c, X = b, Y = a, Z = c
V = c, W = a, X = b, Y = c, Z = a
V = c, W = a, X = b, Y = c, Z = b
V = c, W = a, X = c, Y = b, Z = a
V = c, W = a, X = c, Y = b, Z = c
V = c, W = b, X = a, Y = c, Z = a
V = c, W = b, X = a, Y = c, Z = b
V = c, W = b, X = c, Y = a, Z = b
V = c, W = b, X = c, Y = a, Z = c

3 Problem 3b

$sorted([]).$
$sorted([X]).$

$sorted([X1, X2|T]) :-$
$X1 =< X2,$
$sorted([X2|T]).$

For a list of length 0 and 1, it will always be sorted.
For a list of l+1, where l is a minimum of 1, the sorted method will compare the first two elements, and return false if they are not in order. If they are, it will remove the first element and repeat recursively.
As we have proved that it works for lists with 1 element, and lists of L+1 elements, it is guaranteed to work with lists of all lengths.