Homework

1. Install Mizar, and do the 3 exercises at the end of the file s_0a.miz by adapting the earlier proofs in that file.

   Executing `mizf` on the file will check its correctness and insert messages on this process in the file itself. Alternatively, you can do the same from within emacs (using .emacs)[1] by pressing ctrl-C and enter (if all is well a 'No more errors below!!' message is given).

2. Prove the following in Mizar, for appropriately reserved variables:

```
:: 0x4: {a, {a, b}} = {c, {c, d}} implies a = c & b = d;
:: 0x5: Y c= { x } iff Y = {} or Y = { x };
:: 0x6: Z c= {x1,x2} iff Z = {} or Z = {x1} or Z = {x2} or Z = {x1,x2};
:: 0x7: X <> {x} & X <> {} implies ex y st y in X & y <> x;
```

3. Prove, given that we have

```
reserve A, B, C, D, x, y for set;
reserve V, X, Y, Z, Z1, XX, YY for set;
```

   one problem of your choice of each of the following 4 sets (of 3 problems):

   (a) (binary union)

```
:: 1x1x0: (X \/ Y) \/ Z = X \/ (Y \/ Z);
:: 1x1x1: (X \/ Y) \/ Z = (X \/ Z) \/ (Y \/ Z);
:: 1x1x2: X \/ (X \/ Y) = X \/ Y;
```

   (b) (difference)

```
:: 1x3x7: X \/ (Y \ X) = X \/ Y;
:: 1x3x8: (X \/ Y) \ Y = X \ Y;
:: 1x3x9: (X \ Y) \ Z = X \ (Y \/ Z);
```

   (c) (misses)

```
:: 1x4x14: X \ Y misses Y;
:: 1x4x15: X misses Y implies X misses Y \ Z;
:: 1x4x16: X misses Y \ Z implies Y misses X \ Z;
```

   (d) (symmetric difference)

```
:: 1x5x7: X \/ Y = X \+\ (Y \ X);
:: 1x5x8: (X \+\ Y) \ Z = (X \ (Y \/ Z)) \/ (Y \ (X \/ Z));
:: 1x5x9: X \ Y = X \+\ (X /\ Y);
```

[1] Using this has other benefits such as, e.g. syntax-colouring and showing definitions/results from the library on mouseover (or opening it by shift-clicking).