Unification

> - Two atoms unify only if they are identical.
> - Two structures unify if they have the same functor and arity, **and** if each pair of corresponding arguments unify.
> - A variable unifies with any term, including another variable. As a consequence, the variable is bound with the value of that term.

In the following, indicate which pairs of terms *do not* unify by crossing out the equality sign ($\neq$). If the terms *do* unify, give the variable bindings underneath.

```
a(X, c(d, X)) = a(g, c(d, Y)).
```

```
q(g(X, Y), f(X)) = q(g(h(a), Y), f(h(Z))).
```

```
q(f, Y) = q(Y, X).
```

```
a(b, c, d) = a(X, X, d).
```

```
b(a, X) = b(a, e(Y, c)).
```

```
p(X, g(X), Y) = p(a, g(X), b).
```

```
a(X) = a(b, c).
```

```
p(c, [a,b,c]) = p(X, [Y|Xs]).
```

```
d(c, g(a, e(f, b))) = d(c, g(a, e(f, b))).
```

```
a(b, X) = a(b, c(d, e)).
```

```
q(g(X, Y), f(X)) = q(g(h(Z), Y), f(h(Z))).
```

```
f(y, g(b, A)) = f(Z, g(X, b(X))).
```

```
f(g, h, Y) = f(g, h, f(a, b, f(c, d, e))).
```

```
b(a, X, X) = b(Y, Y, c).
```

```
a(X, Y) = a(b(c, Y), Z).
```

```
p(d, e) = q(X, Y).
```

```
g(f(A), L) = g(X, [X|Xs]).
```

```
p(t(U), t(V), Y) = p(V, R, c).
```

```
p(X, g(X), Y) = p(a, g(X), b).
```

```
a(b, c(d, e(f, g))) = a(b, c(d, e(g, f))).
```

```
p(a, X) = p(X, g(X)).
```

```
c(a, Y, Y) = c(_, _, b).
```

```
p(h(X)) = p(g(X)).
```

```
q(g(X, Y), f(X)) = q(g(h(Z), Y), f(h(Z))).
```

You can check your answers yourself using Prolog on the lab machines:

- Start Prolog in a shell with "`swipl`" (no `&!`)
- Quit with "`halt.`" (note the dot).
- If you make a mistake, type "`abort.`" to come back to the prompt.
- If Prolog answers "`false`", the two terms do not unify.
- If they unify, the variable bindings will be printed.