

4.32 Finding all Solutions to a Goal

findall(+Var, +Goal, -Bag)

Creates a list of the instantiations *Var* gets successively on backtracking over *Goal* and unifies the result with *Bag*. Succeeds with an empty list if *Goal* has no solutions. [findall/3](#) is equivalent to [bagof/3](#) with all free variables bound with the existence operator (\wedge), except that [bagof/3](#) fails when goal has no solutions.

bagof(+Var, +Goal, -Bag)

Unify *Bag* with the alternatives of *Var*, if *Goal* has free variables besides the one sharing with *Var* bagof will backtrack over the alternatives of these free variables, unifying *Bag* with the corresponding alternatives of *Var*. The construct $+Var\wedge Goal$ tells bagof not to bind *Var* in *Goal*. [bagof/3](#) fails if *Goal* has no solutions.

The example below illustrates [bagof/3](#) and the \wedge operator. The variable bindings are printed together on one line to save paper.

```
2 ?- listing(foo).
```

```
foo(a, b, c).
```

```
foo(a, b, d).
```

```
foo(b, c, e).
```

```
foo(b, c, f).
```

```
foo(c, c, g).
```

```
Yes
```

```
3 ?- bagof(C, foo(A, B, C), Cs).
```

```
A = a, B = b, C = G308, Cs = [c, d] ;
```

```
A = b, B = c, C = G308, Cs = [e, f] ;
```

```
A = c, B = c, C = G308, Cs = [g] ;
```

```
No
```

```
4 ?- bagof(C, A^foo(A, B, C), Cs).
```

```
A = G324, B = b, C = G326, Cs = [c, d] ;
```

```
A = G324, B = c, C = G326, Cs = [e, f, g] ;
```

```
No
```

```
5 ?-
```

setof(+Var, +Goal, -Set)

Equivalent to [bagof/3](#), but sorts the result using [sort/2](#) to get a sorted list of alternatives without duplicates.