SWI Prolog
Szukanie wszystkich rozwiązań dla danego celu.

findall(+*Var*,+*Goal*,-*Bag*)
Tworzy listę wszystkich możliwych ukonkretnień zmiennej *Var* przy poszukiwaniu rozwiązania dla celu *Goal*. W przypadku, gdy cel *Goal* nie posiada rozwiązań realizacja findall/3 kończy się sukcesem a wynikiem jest lista pusta.

---

bagof(+*Var*,+*Goal*,-*Bag*)
Unifikuje *Bag* z listą wszystkich rozwiązań alternatywnych dla *Var*. Jeżeli oprócz *Var* cel *Goal* posiada jeszcze inne zmienne wolne, bagof/3 daje rozwiązania alternatywne: dla każdego rozwiązanie dla zmiennych wolnych *Bag* jest unifikowana z listą rozwiązań dla *Var* odpowiadającą temu rozwiązaniu. Użycie konstrukcji +*Var*^*Goal* zamiast +*Goal* powoduje, że zmienna *Var* pozostanie wolna w celu *Goal*.
Jeżeli wszystkie zmienne w celu *Goal* oprócz zmiennej związanej z pierwszym argumentem bagof/3 występują z operatorem (^) i istnieje rozwiązanie celu *Goal* dla *Var*, to użycie bagof/3 pokrywa się z użyciem predykatu findall/3.
Uwaga! Jeżeli nie ma rozwiązania dla celu *Goal*, to próba realizacji bagof/3 kończy się porażką.

---

setof(+Var,+Goal,-Set)
Równoważny z bagof/3, z tym że lista rozwiązań jest posortowaną listą bez powtórzeń.

Przykłady użycia predykatów findall/3, bagof/3, setof/3

Dany jest program zawierający definicję predykatu foo/3:

```
foo(a, b, c).
foo(a, b, d).
foo(b, c, e).
foo(b, c, f).
foo(c, c, c).
```

Predykat findall/3:

Cel:

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

oznacza, że szukamy wszystkich rozwiązań dla zmiennej C niezależnie od wartości pozostałych zmiennych występujących w celu.
W tym przypadku otrzymujemy jedno rozwiązanie w postaci listy Cs wszystkich rozwiązań dla zmiennej C niezależnie od wartości zmiennych A i B.

```
CS = [c, d, e, f, c].
```

Wszystkie rozwiązania celu

```
?- foo(A,B,C).
```

to

```
A = a, B = b, C = c ;
A = a, B = b, C = d ;
A = b, B = c, C = e ;
A = b, B = c, C = f ;
A = c, B = c, C = c.
```

Na liście Cs umieszczono tylko rozwiązania dla C.

Realizacja celu: ?-foo(d,B,C) kończy się porażką

```
?- foo(d,B,C).
false.
```

Jednak realizacja celu

```
?- findall(C,foo(d,B,C),Cs).
```

kończy się sukcesem i w wyniku otrzymujemy pustą listę rozwiązań dla zmiennej C.

```
Cs
```

Predykat bagof/3:

Dla celu

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

Otrzymujemy listę Cs wszystkich wartości zmiennej C, dla ustalonych wartości zmiennych A i B. Mamy wtedy rozwiązania alternatywne.

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

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

A = c
B = c
Cs = [c].
```

Wszystkie rozwiązania celu

```
?- foo(A,B,C).
```

to

```
A = a, B = b, C = c ;
A = a, B = b, C = d ;
```

Dla wartości A=a i B=b mamy dwie wartości dla C, stąd Cs=[c,d].

```
A = b, B = c, C = e ;
A = b, B = c, C = f ;
```

Dla wartości A=b i B=c mamy dwie wartości dla C, stąd Cs=[e,f]

```
A = c, B = c, C = c.
```

Dla wartości A=c i B=c mamy tylko jedną wartość dla zmiennej C. Stąd Cs=[c].

Cel ?-foo(d,B,C) kończy się porażką. Wtedy również cel

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

```
false
```

kończy się porażką.

Cel:

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

Oznacza, że szukamy wszystkich rozwiązań dla zmiennej C, przy ustalonej wartości zmiennej B, ale niezależnie od wartości zmiennej A.

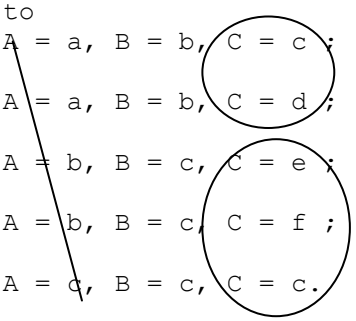
```
B = b
Cs = [c, d] ;

B = c
Cs = [e, f, c] .
```

Wszystkie rozwiązania celu

```
?- foo(A,B,C).
```

to

```
A = a, B = b, C = c ;
A = a, B = b, C = d ;
A = b, B = c, C = e ;
A = b, B = c, C = f ;
A = c, B = c, C = c.
```



Dla wartości B=b, niezależnie od wartości zmiennej A, mamy dwie wartości dla C, stąd Cs=[c,d]

Dla wartości B=c, niezależnie od wartości zmiennej A, mamy dwie wartości dla C, stąd Cs=[e,f,c]

Cel

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

Oznacza, że szukamy wszystkich rozwiązań dla C niezależnie od wszystkich pozostałych zmiennych. Cel ten daje taki sam wynik jak ?-findall((C,foo(A,B,C),Cs)

```
Cs = [c, d, e, f, c].
```

Predykat setof/3:

Działa analogicznie jak bagof/3, z tym że otrzymujemy uporządkowaną listę rozwiązań bez powtórzeń

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

```
Cs = [c, d, e, f].
```