

1 Modes

We often talk about modes in Prolog. Modes are a way of describing how a predicate will be used. We denote an argument of a predicate with + if that argument is provided to the predicate, and with - if that argument is outputted by the predicate. A defined Prolog predicate can often work with many different modes.

As a small example, recall the `color` predicate from the homework. There are two possible modes for this predicate: `color(+col)` and `color(-col)`. The predicate works correctly with both modes. If we try `color(red)` or `color(blue)`, we get confirmation that those satisfy the predicate (as we defined them to). If we try `color(C)` in Prolog, we will receive all possible colors we defined (there are four). In general, the mode of a predicate which modes all arguments positively is always just a verifier that some instance satisfies the predicate.

Note that not all possible modes work correctly for a given predicate in Prolog. Many predicates do not work with all arguments moded negatively. For example, given a predicate `permutation(L, P)`, where `L` is a list and `P` is a permutation of `L`, the mode `permutation(-L, -P)` will not terminate because there are infinite pairs (L, P) such that `P` is a permutation of `L`.

Task 1. Let the predicate `zip` be defined as follows:

```
zip([], [], []).  
zip([X|L], [Y|M], [(X, Y)|P]) :- zip(L, M, P).
```

Which modes does `zip` work correctly with?

Solution 1: There are 5 modes that work correctly; the other 3 do not terminate.

- (+, +, +)
- (+, +, -)
- (-, +, +)
- (+, -, +)
- (-, -, +)

Task 2. Let the predicate `mult` be defined as follows:

```
nat(z).  
nat(s(N)) :- nat(N).
```

```
plus(z, N, N).  
plus(s(M), N, s(P)) :- plus(M, N, P).
```

```
mult(z, N, z).  
mult(s(M), N, P) :- plus(Q, N, P), mult(M, N, Q).
```

Which modes does `mult` work correctly with?

Solution 2: Note that `(+, +, -)` does not work, even though it might be expected to, due to the infinite possibilities in the `plus(-, +, -)` mode.

- `(+, +, +)`
- `(+, -, +)`
- `(-, +, +)`

Task 3. How might we rewrite `mult` if we wanted it to work correctly with the `mult(+, +, -)` modality?

Solution 3: Change the order of the `plus` and `mult` in the third clause of the predicate.