CSE 341 - Programming Languages - Autumn 2012

The definition of "append" in the lecture notes is as follows:

```
myappend([],Ys,Ys).
myappend([X|Xs],Ys,[X|Zs]) :- myappend(Xs,Ys,Zs).
```

If you're more used to an append function, this may look confusing. Here is another version:

```
myappend([],Ys,Ys).
myappend([X|Xs],Ys,Answer) :- myappend(Xs,Ys,Zs), Answer=[X|Zs].
```

The two versions work the same most of the time, but the first one is the standard way to write this rule in Prolog, and is considered better style. It also has a stronger implication that this is the append *relation* rather than the append *function*.

Also, there are cases where the nonstandard rule gets stuck in an infinite search, while the standard one works OK. If you try this goal:

```
?- myappend(A,B,[1,2])
```

and backtrack through all the answers, the standard rule will give 3 answers and fail, while the non-standard one will give the same 3 answers and then get stuck in an infinite search. (Try tracing it to see why.)