3. 1 unify[t(s(s), G, s, p, t(K), s), t(s(G), G, s, p, t(K), U)].

Init: sub = {}, equations: [(t(s(s), G, s, p, t(K), s) = t(s(G), G, s, p, t(K), U)]

Both sides are atomic, same symbol predicate and same number of terms, so we split into more equations.

.  
next iteration;  
Both sides are atomic, same symbol predicate and same number of terms, so we split into more equations.

.  
next iteration;  
Same variable on both sides, we continue to the next iteration.  
same Constant symbol, continue to next iteration.  
same Constant symbol, continue to next iteration.  
Both sides are atomic, same symbol predicate and same number of terms, so we split into more equations.  
.  
We have a variable on one side, constant on the other:  
next iteration;

We have a variable on one side, constant on the other:  
next iteration;  
Same variable on both sides, we continue to the next iteration.  
equations is empty, we return .  
We can confirm:  
 .  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
unify[p([v | [V | W]]), p([[v | V] | W])]

Init:

Both sides are atomic, same symbol predicate and same number of terms, so we split into more equations.

Equations=[ ([ v|[V|W]]=[[v|V]|W]) ][ ([ v|[V|W]]=[[v|V]|W]) ]

Next iteration;

We split the equations :

Next iteration;

Now we can see that on the left side we have a constant, while in the right side we have a predicate with binary arity. Hence, the algorithm will terminate and return FAIL.