```
Trytomakework(p,temp)
  execute temp unless we know from memo that temp will fail
  if the execution of temp fails
    Find the first call, say from Pi.vi’ to Pj.vj’,that fails
    Record in memo that Pi.vi’ calling Pj.vj’ fails
    if Pi == p or Pi is earlier than p in todolist then
        possible(Pi) = {vi’}
    if Pj == p or Pj is earlier than p in todolist then
        possible(Pj) = {vj’}
    if neither Pi nor Pj belongs to the set {x | x = p
                        or a package earlier than p in the todolist}
        possible(Pi) = {all versions of Pi in sourcemap}
        possible(Pj) = {all versions of Pj in sourcemap}


     for each configuration c that can be constructed from the cross-product of possible(Pi) and possible(Pj) such that the version vi’’ of Pi and vj’’ of Pj has the property that (Pi.vi’’,Pj.vj’’) is not in memo
          ret = trytomakework(temp)
          if (ret is not null) then return ret
     end for
return null
else return temp # this configuration works
```

The condition is not right as Possible(pj) (resp Possible(Pi) will never be filled if Pi=p (resp Pj=p) + to avoid infinite “recursive calls” we should limit versions in sourcemap to versions > vi’ (resp vj’).

Just missing p in parameters

Proposed reformulation:

```
Trytomakework(p,temp)
  execute temp unless we know from memo that temp will fail
  if the execution of temp fails
    Find the first call, say from Pi.vi’ to Pj.vj’,that fails
    Record in memo that Pi.vi’ calling Pj.vj’ fails
    if Pi == p or Pi is earlier than p in todolist
        then possible(Pi) = {vi’}
        else possible(Pi) = {all versions of Pi in sourcemap >= vi}
    if Pj == p or Pj is earlier than p in todolist
        then possible(Pj) = {vj’}
        else possible(Pj) = {all versions of Pj in sourcemap >= vj’}
     for each configuration c that can be constructed from the cross-product of possible(Pi) and possible(Pj) such that the version vi’’ of Pi and vj’’ of Pj has the property that (Pi.vi’’,Pj.vj’’) is not in memo
          ret = trytomakework(p, temp)
          if (ret is not null) then return ret
     end for
return null
else return temp # this configuration works
```