trytomakework(p,temp)

```
    execute temp unless we know from memo that temp will fail
    while 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 equals p or a package earlier in the todolist then
            possible(Pi) = {all versions of Pi in sourcemap}
            possible(Pj) = {all versions of Pj in sourcemap}
        The condition will not work (Pi nor Pj, if only one of these is true the other “possible” list
        should be filled). ➔ Easy fix: use an else on the first conditions.


        Form a new version of temp with some vi’’ of Pi and version vj’’ of Pj
            from the cross-product of possible(Pi)
            and possible(Pj) such that (Pi.vi’’,Pj.vj’’) is not in memo
            if there is no such version then return null
            execute temp
        ⇨   Complexity will be much higher and we still have no guarantee that it will end using
            the example I sent you.


    endwhile
    return temp # this configuration works
```