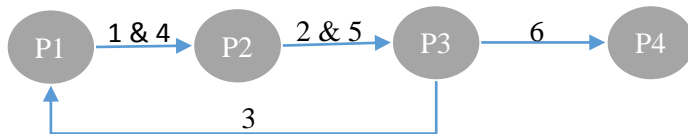


Example that fails:

Inputs:

Packages: P1, P2, P3, P4

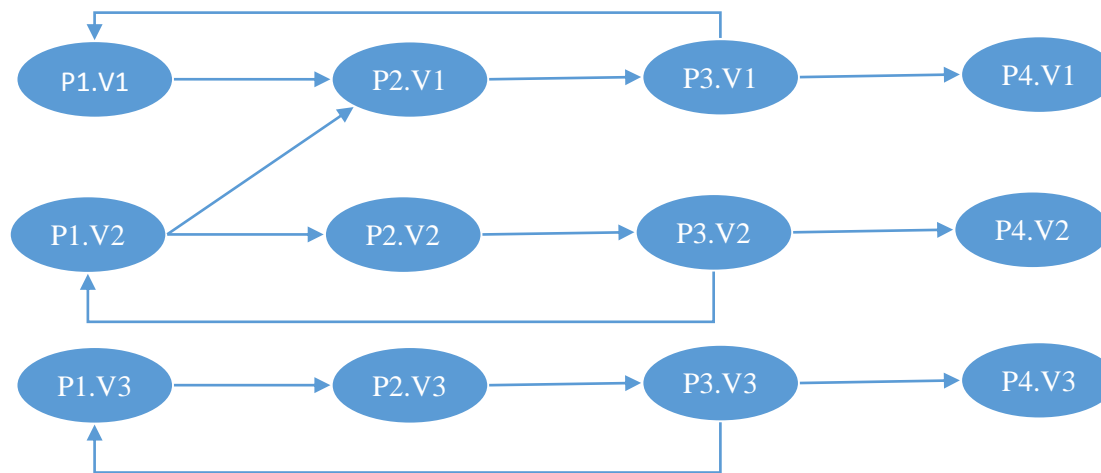
Call dependencies:



Nb: the numbers on the arrows represent the calling order.

Basically an execution will look like P1->P2->P3->P1-P2->P3->P4.

Versions Compatibilities:



Initial configuration: P1.V1/P2.V1/P3.V1/P4.V1

Query: Q(P4)

Execution :

Phase/Comments	Variables	MEMO
Init	Current= P1.V1/P2.V1/P3.V1/P4.V1 todoList=P1 constaints=∅ sourceMap={{[P1,(V1,V2,V3)], [P2,(V1,V2,V3)], [P3,(V1,V2,V3)], [P4,(V1,V2,V3)]}}	∅
Main For p in todolist in desc order	p=P4	
Update current to the highest p that works ...	Current= P1.V1/P2.V1/P3.V1/P4.V1	
If current has less than last version of p... versionsTodo=...	versionsTodo=(P4.V2, P4.V3)	
For each v in versionsTodo ... Temp := current with p set to v Ret:=tryToMakeWork (p, temp)	Temp = P1.V1/P2.V1/P3.V1/ P4.V2	
1.0 tryToMakeWork (P4, Temp)		
Find the first call, say from Pi.vi' to Pj.vj',that fails Record in memo that Pi.vi' calling Pj.vj' fails	Pi.vi' = P3.V1 Pj.vj' = P4.V2	+ (P3.V1, P4.V2)
if Pi == p or Pi is earlier than p in todolist then possible(Pi) = {vi'} else possible(Pi) = {all versions of Pi in sourcemap} if Pj == p or Pj is earlier than p in todolist then possible(Pj) = {vj'} else possible(Pj) = {all versions of Pj in sourcemap}	Possible(Pi) = P3.V1, P3.V2, P3.V3 Possible(Pj) = P4.V2	
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)	C1.1 =(P1.V1,P2.V1,P3.V2, P4.V2) C1.2 =(P1.V1,P2.V1,P3.V3, P4.V2)	
1.1 tryToMakeWork (P4, [P1.V1/P2.V1/P3.V2/P4.V2])	Temp = [P1.V1/P2.V1/P3.V2/P4.V2]	
Find the first call, say from Pi.vi' to Pj.vj',that fails Record in memo that Pi.vi' calling Pj.vj' fails	Pi.vi' = P2.V1 Pj.vj' = P3.V2	+ (P2.V1, P3.V2)
if Pi == p or Pi is earlier than p in todolist then possible(Pi) = {vi'} else possible(Pi) = {all versions of Pi in sourcemap} if Pj == p or Pj is earlier than p in todolist then possible(Pj) = {vj'}	Possible(Pi) = P2.V1, P2.V2, P2.V3 Possible(Pj) = P3.V1, P3.V2, P3.V3	

<p>else possible(Pj) = {all versions of Pj in sourcemap}</p> <p>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)</p>	<p>C1.1.1 =(P1.V1,P2.V1,P3.V1, P4.V2) C1.1.2 =(P1.V1,P2.V2,P3.V1, P4.V2) C1.1.3 =(P1.V1,P2.V3,P3.V1, P4.V2) C1.1.4 =(P1.V1,P2.V2,P3.V2, P4.V2) C1.1.5 =(P1.V1,P2.V3,P3.V2, P4.V2) C1.1.6 =(P1.V1,P2.V1,P3.V3, P4.V2) C1.1.7 =(P1.V1,P2.V2,P3.V3, P4.V2) C1.1.8 =(P1.V1,P2.V3,P3.V3, P4.V2)</p>	<p>Would do infinite loop > limit to versions upper than vi (resp vj) for possible(pi) (resp Possible(pj))</p>
	-	

