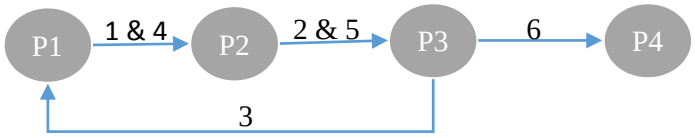
Example that fails:

Inputs:

Packages: P1, P2, P3, P4

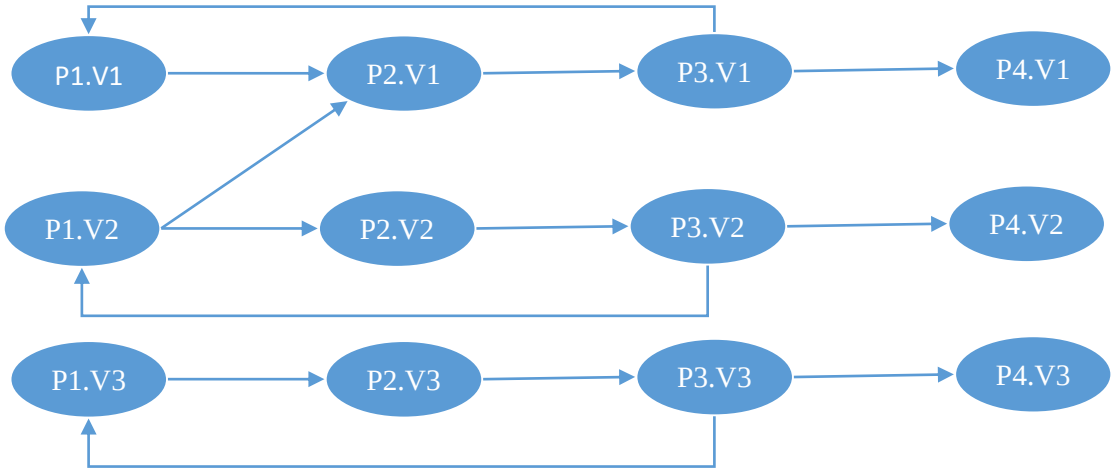
Call dependencies:



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=ø sourceMap={[P1,(V1,V2,V3)], [P2,(V1,V2,V3)], [P3,(V1,V2,V3)], [P4,(V1,V2,V3)]}	ø
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**	
tryToMakeWork (P4, Temp)	initalTemp= P1.V1/P2.V1/P3.V1/P4.V2	
While temp doesn't work …		
Iteration1		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P3.V1 & Pj.vj'=P4.V2	+ (P3.V1 →P4.V2)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and …	Temp = P1.V1/P2.V1/P3.V1/**P4.V3**	
Iteration2		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P3.V1 & Pj.vj'=P4.V3	+ (P3.V1 →P4.V3)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and for which if ***there is no such version vj''*** then **advance Pi within temp to the next version vi''** **and start Pj from the version in initialtemp** …	Temp = P1.V1/P2.V1/**P3.V2**/**P4.V2**	
Iteration3		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P2.V1 & Pj.vj'=P3.V2	+ (P2.V1 →P3.V2)

Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and …	Temp = P1.V1/P2.V1/**P3.V3**/P4.V2	
Iteration4		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P2.V1 & Pj.vj'=P3.V3	+ (P2.V1 →P3.V3)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and for which if ***there is no such version vj''*** then **advance Pi within temp to the next version vi''** **and start Pj from the version in initialtemp** …	Temp = P1.V1/**P2.V2**/**P3.V1**/P4.V2	
Iteration5		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P1.V1 & Pj.vj'=P2.V2	+ (P1.V1 →P2.V2)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and …	Temp = P1.V1/**P2.V3**/P3.V1/P4.V2	
Iteration6		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P1.V1 & Pj.vj'=P2.V3	+ (P1.V1 →P2.V3)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and for which if ***there is no such version vj''*** then **advance Pi within temp to the next version vi''** **and start Pj from the version in initialtemp** …	Temp = **P1.V2**/**P2.V1**/P3.V1/P4.V2	
Iteration7		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P3.V1 & Pj.vj'=P1.V2	+ (P3.V1 →P1.V2)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and …	Temp = **P1.V3**/P2.V1/P3.V3/P4.V2	
Iteration8		
Find first call Pi.vi' to Pj.vj' that fails	Pi.vi' = P3.V1 & Pj.vj'=P1.V3	

Record in memo that it fails		+ (P3.V1 →P1.V3)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and for which if ***there is no such version vj''*** then **advance Pi within temp to the next version vi''** **and start Pj from the version in initialtemp** …	Temp = **P1.V1**/P2.V1/**P3.V2**/P4.V2 ⇨ **Thats where it fails, we will go back to iteration 3 for an infinite loop**	

Other example (that eventually works but too many tests):

Inputs :

Packages: P1, P2, P3, P4

Call dependencies: P4→P3→P2→P1

Compatibilities:

- P1.V1/P2.V1/P3.V1/P4.V1
- P1.V1/P2.V1/P3.V2/P4.V2
- P1.V2/P2.V2/P3.V2/P4.V2
- P1.V3/P2.V3/P3.V3/P4.V3

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

Query: Q(P1)

Execution :

Phase/Comments	Variables	MEMO
Init	Current= P1.V1/P2.V1/P3.V1/P4.V1 todoList=P1 constaints=ø sourceMap={[P1,(V1,V2,V3)], [P2,(V1,V2,V3)], [P3,(V1,V2,V3)], [P4,(V1,V2,V3)]}	ø
Main For p in todolist in desc order	p=P1	
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=(P1.V2, P1.V3)	
For each v in versionsTodo … Temp := current with p set to v Ret:=tryToMakeWork (p, temp)	Temp = P1.V2/P2.V1/P3.V1/P4.V1	

tryToMakeWork (P1, Temp)	initalTemp= P1.V2/P2.V1/P3.V1/P4.V1	
While temp doesn't work …		
Iteration1		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P2.V1 & Pj.vj'=P1.V2	+ (P2.V1 →P1.V2)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and …	Temp = **P1.V3**/P2.V1/P3.V1/P4.V1	
Iteration2		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P2.V1 & Pj.vj'=P1.V3	+ (P2.V1 →P1.V3)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and for which if ***there is no such version vj''*** then **advance Pi within temp to the next version vi''** **and start Pj from the version in initialtemp** …	Temp = **P1.V2**/**P2.V2**/P3.V1/P4.V1	
Iteration3		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P3.V1 & Pj.vj'=P2.V2	+ (P3.V1 →P2.V2)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj …	Temp = P1.V2/**P2.V3**/P3.V1/P4.V1	
Iteration4		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P3.V1 & Pj.vj'=P2.V3	+ (P3.V1 →P2.V3)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and for which if ***there is no such version vj''*** then **advance Pi within temp to the next version vi''** **and start Pj from the version in initialtemp** …	Temp = P1.V2/**P2.V1**/**P3.V2**/P4.V1 This is where it fails as reverting to the initialTemp reverts P2 to V1 while it should revert only to P2.V2 which was validated in iteration 2.	
Iteration5		

Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P4.V1 & Pj.vj'=P3.V2	+ (P4.V1 →P3.V2)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj …	Temp = P1.V2/P2.V1/**P3.V3**/P4.V1	
Iteration6		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P4.V1 & Pj.vj'=P3.V3	+ (P4.V1 →P3.V3)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and for which if ***there is no such version vj''*** then **advance Pi within temp to the next version vi''** **and start Pj from the version in initialtemp** …	Temp = P1.V2/P2.V1/**P3.V1**/**P4.V2** **Here again we restore P3 to V1 while it should stay in V2.**	
Iteration7		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P4.V2 & Pj.vj'=P3.V1	+ (P4.V2 →P3.V1)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj …	Temp = P1.V2/P2.V1/**P3.V2**/P4.V2	
Iteration8		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P2.V1 & Pj.vj'=P1.V2	*Already in memo*
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj …	Temp = **P1.V3**/P2.V1/P3.V2/P4.V2	
Iteration9		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P2.V1 & Pj.vj'=P1.V3	Already in memo
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and for which if ***there is no such version vj''*** then	Temp = **P1.V2/P2.V2**/P3.V2/P4.V2	

advance Pi within temp to the next version vi'' **and start Pj from the version in initialtemp** …		

Initial example (that will eventually work although not optimized):

Inputs :

Packages: P1, P2, P3, P4

Call dependencies: P4→P3→P2→P1

Compatibilities:

- P1.V1/P2.V1/P3.V1/P4.V1
- P1.V2/P2.V2/P3.V2/P4.V2
- P1.V3/P2.V3/P3.V3/P4.V3

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

Query: Q(P1)

Execution :

Phase/Comments	Variables	MEMO
Init	Current= P1.V1/P2.V1/P3.V1/P4.V1 todoList=P1 constaints=ø sourceMap={[P1,(V1,V2,V3)], [P2,(V1,V2,V3)], [P3,(V1,V2,V3)], [P4,(V1,V2,V3)]}	ø
Main For p in todolist in desc order	p=P1	
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=(P1.V2, P1.V3)	
For each v in versionsTodo … Temp := current with p set to v Ret:=tryToMakeWork (p, temp)	Temp = P1.V2/P2.V1/P3.V1/P4.V1	
tryToMakeWork (P1, Temp)	initalTemp= P1.V2/P2.V1/P3.V1/P4.V1	

While temp doesn't work …		
Iteration1		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P2.V1 & Pj.vj'=P1.V2	+ (P2.V1 →P1.V2)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and …	Temp = **P1.V3**/P2.V1/P3.V1/P4.V1	
Iteration2		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P2.V1 & Pj.vj'=P1.V3	+ (P2.V1 →P1.V3)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and for which if ***there is no such version vj''*** then **advance Pi within temp to the next version vi''** **and start Pj from the version in initialtemp** …	Temp = **P1.V2**/**P2.V2**/P3.V1/P4.V1	
Iteration3		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P3.V1 & Pj.vj'=P2.V2	+ (P3.V1 →P2.V2)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj …	Temp = P1.V2/**P2.V3**/P3.V1/P4.V1	
Iteration4		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P3.V1 & Pj.vj'=P2.V3	+ (P3.V1 →P2.V3)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and for which if ***there is no such version vj''*** then **advance Pi within temp to the next version vi''** **and start Pj from the version in initialtemp** …	Temp = P1.V2/**P2.V1**/**P3.V2**/P4.V1 This is where it fails as reverting to the initialTemp reverts P2 to V1 while it should revert only to P2.V2 which was validated in iteration 2.	
Iteration5		
Find first call Pi.vi' to Pj.vj' that fails	Pi.vi' = P4.V1 & Pj.vj'=P3.V2	

Record in memo that it fails		+ (P4.V1 →P3.V2)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj …	Temp = P1.V2/P2.V1/**P3.V3**/P4.V1	
Iteration6		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P4.V1 & Pj.vj'=P3.V3	+ (P4.V1 →P3.V3)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj and for which if ***there is no such version vj''*** then **advance Pi within temp to the next version vi''** **and start Pj from the version in initialtemp** …	Temp = P1.V2/P2.V1/**P3.V1**/**P4.V2** **Here again we restore P3 to V1 while it should stay in V2.**	
Iteration7		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P4.V2 & Pj.vj'=P3.V1	+ (P4.V2 →P3.V1)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj …	Temp = P1.V2/P2.V1/**P3.V2**/P4.V2	
Iteration8		
Find first call Pi.vi' to Pj.vj' that fails Record in memo that it fails	Pi.vi' = P3.V2 & Pj.vj'=P2.V1	+ (P3.V2 →P2.V1)
Form a new version of temp with Pj.vj'' which is the next version above vj' in Pj …	Temp = P1.V2/**P2.V2**/P3.V2/P4.V2	