

3.1

Al tener ciclos en un grafo de amistades, el solver se queda atrapado buscando cadenas infinitamente. Para resolver este problema con el operador `#count`, tenemos varias alternativas. Se propone utilizar el operador `#count` para contar la cantidad de amistades que hay en el grafo y así verificar que el largo de la cadena sea menor o igual a este valor. De esta forma, no se evitará 100% el problema de contar más de una vez una cadena con largos distintos dado que hay loops entremedio, pero se limita la cantidad de cadenas para que el solver no se enrede con los ciclos y se asegura obtener todos los pares de usuarios que están conectados en una cadena. En concreto se propone que el predicado “cadena/3” tenga la siguiente forma:

```
cadena(X,Y,Largo) :- amistad(X,Y), Largo=1.  
cadena(X,Y,Largo) :- cadena(X,Z,L1), cadena(Z,Y,L2), Largo=L1+L2, #count{A,B : amistad(A,B)} >= Largo, X!=Y.
```

(Se agregó además la restricción de X distinto de Y)