

Introducción a la Programación

Segundo Entregable de laboratorio

Ejercicio 5

Contamos con un listado de vuelos, donde cada vuelo está representado por dos datos: la ciudad de la cual parte el vuelo (origen), y la ciudad a la cual llega (destino). Implementar la función **sePuedeLlegar**, que indica si hay una forma de llegar desde una ciudad a otra utilizando las conexiones disponibles en el listado de vuelos, y en caso afirmativo devuelve la cantidad de vuelos que deben tomarse.

La especificación detallada de la función que debe implementarse es la siguiente:

```
problema sePuedeLlegar (in origen: String, in destino: String, in vuelos: seq⟨String × String⟩) : ℤ {
  requiere: {origen ≠ destino}
  requiere: {soloParteUnVueloDeCadaCiudad(vuelos) ∧ soloLlegaUnVueloAcadaCiudad(vuelos) ∧ sinRepetidos(vuelos)}
  asegura: {hayRuta(vuelos, origen, destino) → largoDeRuta(vuelos, origen, destino, res)}
  asegura: {¬hayRuta(vuelos, origen, destino) → res = -1}
}

pred soloParteUnVueloDeCadaCiudad (vuelos: seq⟨String × String⟩) {
  (∀i, j : ℤ)((0 ≤ i < |vuelos| ∧ 0 ≤ j < |vuelos| ∧ i ≠ j) → vuelos[i]0 ≠ vuelos[j]0)
}

pred soloLlegaUnVueloAcadaCiudad (vuelos: seq⟨String × String⟩) {
  (∀i, j : ℤ)((0 ≤ i < |vuelos| ∧ 0 ≤ j < |vuelos| ∧ i ≠ j) → vuelos[i]1 ≠ vuelos[j]1)
}

pred hayRuta (vuelos: seq⟨String × String⟩, origen, destino: String) {
  (∃ruta : seq⟨String × String⟩)(vuelosValidos(ruta, vuelos) ∧ |ruta| ≥ 1 ∧ ruta[0]0 = origen ∧ ruta[|ruta| - 1]1 =
  destino ∧ caminoDeVuelos(ruta))
}

pred vuelosValidos (ruta, vuelos: seq⟨String × String⟩) {
  sinRepetidos(ruta) ∧ (∀v : String × String)(v ∈ ruta → v ∈ vuelos)
}

pred caminoDeVuelos (vuelos: seq⟨String × String⟩) {
  (∀i : ℤ)(1 ≤ i < |ruta| → ruta[i]0 = ruta[i - 1]1)
}

pred largoDeRuta (ruta: seq⟨String × String⟩, origen, destino: String, longCamino: ℤ) {
  (∃ruta : seq⟨String × String⟩)(vuelosValidos(ruta, vuelos) ∧ |ruta| ≥ 1 ∧ ruta[0]0 = origen ∧ ruta[|ruta| - 1]1 =
  destino ∧ caminoDeVuelos(ruta) ∧ |ruta| = longCamino)
}
```

Nota: Si desean realizar pruebas, se espera que el input tenga el siguiente formato:

origen

destino

ciudadOrigen [COMA] ciudadDestino [ESPACIO] ciudadOrigen [COMA] ciudadDestino...

Por ejemplo:

rosario jujuy misiones,jujuy salta,chubut rosario,misiones
--

Notar que cada vuelo se separa del siguiente mediante un espacio, y las ciudades origen y destino de un vuelo se separan por una coma

En este caso, el origen sería *Rosario*, el destino *Jujuy*, y existen 3 vuelos (*Misiones* → *Jujuy*, *Salta* → *Chubut* y *Rosario* → *Misiones*)