Problema 1

(a)

Tengo que especificar el predicado: esHanoiValido (v:vector<vector<int>>).

1. Necesito que $N > 2$ y que $K > 2$. Son las condiciones básicas del juego. Predicado: *condicionesMinimas*
2. Todos los discos tienen que estar ordenados en orden descendente en todos los palos. Si hay un sólo disco en un palo, se asume que está ordenado; y si no hay ninguno, también. Predicado: *ordenados*
3. Si tengo K palos, entonces debe haber K palos en el juego. Es decir, debe haber K vectores dentro del vector exterior. Predicado: *K_palosPresentes*
4. Si tengo N discos, quiere decir que el cardinal de todos los palos debe ser igual a N. Todos los discos están en algún palo. Predicado: *todosPresentes*
5. Todos los discos deben tener un diámetro distinto. No hay diámetros repetidos. Predicado: *diametrosDistintos*
6. Cada palo puede tener a lo sumo N discos, ni uno más. Predicado: *capacidadMaximaN*

esHanoiValido (v:vector<vector<int>>) $\equiv$
condicionesMinimas (v) $\wedge$
$(\forall i{:}\text{int})(0 \leq i < |v| \Rightarrow$ ordenados $(v)) \wedge$
Defino predicados auxiliares:

- condicionesMinimas (v) $\equiv$
  tiene3Palos (v) $\wedge$ tiene3Discos (v)
  - tiene3Palos (v:vector<vector<int>>) $\equiv (|v| \geq 3)$
    El tamaño debe ser mayor o igual a 3.
  - tiene3Discos (v:vector<vector<int>>) $\equiv (\forall i{:}\text{int})(0 \leq i < |v| \Rightarrow \sum_{j=0}^{|v[i]|-1} v[i][j] \geq 3)$
    La suma de todos los discos en los palos debe ser mayor o igual a tres.
- ordenados (v) $\equiv$
  $(\forall i{:}\text{int}0 \leq i < |v| \Rightarrow$ hayUnDisco $(v[i]) \vee$ noHayDiscos $(v[i]) \vee$ ordenDec $(v[i]))$
  $\wedge$ noHayRepetidos (v)
  - hayUnDisco (vec:vector<int>) $\equiv (|vec| = 1)$
    Si hay un disco en algún palo, se asume que está ordenado.
  - noHayDiscos (vec:vector<int>) $\equiv (|vec| = 0)$
    Si no hay discos en un palo, se asume que está ordenado también.
  - ordenDec (vec:vector<int>) $\equiv (|vec| \geq 2) \wedge (\forall k{:}\text{int})(0 \leq k < |vec| - 1 \Rightarrow v[k] \leq v[k+1])$
    Si un palo tuviese dos discos, queremos que estén ubicados en orden descendente.
  - noHayRepetidos (v:vector<vector<int>>) $\equiv (\forall i{:}\text{int})(0 \leq i < |v| \wedge |v[i]| \neq 0 \Rightarrow (\forall j{:}\text{int})(0 \leq j \leq |v[i]| - 1 \Rightarrow v[i][j] \neq v[i][j+1]))$
    Si un palo está vacío o con un elemento no lo tenemos en consideración, pero tiene más de un elemento queremos que no aparezca más de una vez en ningún palo.

Problema 2

(a)

- $P_c \equiv (|vec| \text{ mód } 2 = 0) \wedge \text{ningunoVacío }(\text{vec}) \wedge (suma = 0) \wedge (i = |vec|)$
  Por la precondición de la función, el tamaño de *vec* debe ser par y no puede haber strings vacíos. Por la precondición del bucle, *suma* se inicializa en cero e i es el tamaño de *vec*.

  ningunoVacío (v:vector<string>) $\equiv (\forall i\text{:int})(0 \leq i < |v| \Rightarrow |v[i]| > 0)$

- $Q_c \equiv (i = 0) \wedge (suma = \sum_{i=0}^{|vec|-1} \beta(vec[i][0] = vec[i][|vec| - 1]) + \beta(vec[i+1][0] = vec[i+1][|vec| - 1]))$
  i terminará valiendo 0 porque se necesita que se niegue la guarda. Básicamente, lo que hace el ciclo, es que para cada índice j, se chequean si cumple ese índice y el índice próxim, como es $j + 1$. Por eso, la sumatoria se fija si cumple un índice o el siguiente (pueden ser los dos).

(b)

- $\mathcal{I} \equiv (0 \leq i \leq |vec|) \wedge (suma = \sum_{j=0}^{i-1} \beta(vec[j][0] = vec[j][|vec| - 1]) + \beta(vec[j+1][0] = vec[j+1][|vec| - 1]))$
  Como sabemos que $|vec|$ mód $2 = 0$, i terminará valiendo cero.

(c)

- Cumple con el invariante: $i = 2, |vec| = 6, vec = \{\text{‘}aa', \text{‘}ba', \text{‘}bb', \text{‘}cd', \text{‘}dd', \text{‘}ff'\}, suma = 3$
- No cumple con el invariante: $i = 2, |vec| = 5, vec = \{\text{‘}ba', \text{‘}bb', \text{‘}cd', \text{‘}dd', \text{‘}ff'\}, suma = 2$
  $|vec|$ no puede ser impar, y está mal *suma*, debería ser 3.

(d)

Me piden demostrar que:

$$\mathcal{I} \wedge \neg B \Rightarrow Q_c$$

Esto es,

$$\left((0 \leq i \leq |vec|) \wedge (suma = \sum_{j=0}^{i-1} \beta(vec[j][0] = vec[j][|vec| - 1]) + \beta(vec[j+1][0] = vec[j+1][|vec| - 1]))\right) \wedge (i \leq 1)$$

$$\Rightarrow \left((i = 0) \wedge (suma = \sum_{i=0}^{|vec|-1} \beta(vec[i][0] = vec[i][|vec| - 1]) + \beta(vec[i+1][0] = vec[i+1][|vec| - 1]))\right)$$

$Q_c : (i = 0)$ vale, porque por $\mathcal{I} : (0 \leq i \leq |vec|)$ y $\neg B : (i \leq 1)$ concluimos que $0 \leq i \leq 1 \Rightarrow i : [0, 1]$
$Q_c\ :\ (\Big((i = 0) \wedge (suma = \sum_{i=0}^{|vec|-1} \beta(vec[i][0] = vec[i][|vec| - 1]) + \beta(vec[i+1][0] = vec[i+1][|vec| - 1]))\Big))$
vale, porque se cumple que de $\mathcal{I} : (suma = \ldots)$ da el mismo resultado si $i = 0$ ó si $i = 1$.