

Práctica 1 – Ejercicios

Nuestro primer objetivo es conseguir que MiniZinc esté accesible y funcione.

Linux

1. Abrir un terminal (arriba a la derecha; “Actividades” y luego “terminal”; si ya hay un terminal y queremos otro, dar botón derecho sobre el terminal y elegir “Ventana Nueva”)
2. Tecleamos *minizinc*. Si la respuesta es algo del estilo

```
minizinc: no model file specified
minizinc: use --help for more information
```

es que todo va bien; en otro caso avisar al profesor

Windows

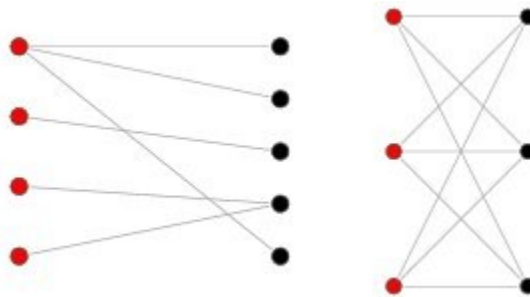
En el símbolo de inicio de Windows buscamos MiniZinc IDE

Es posible que la primera vez nos aparezca “Could not find mzn2fzn executable”.

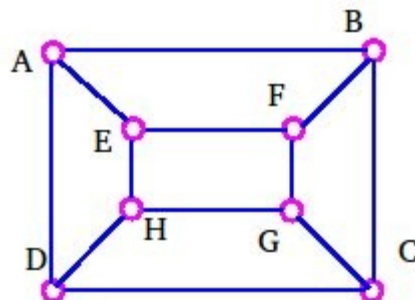
En ese caso:

1. haremos click en el boton “Select” de la opción MiniZinc Path de la ventana que nos aparece (la ventana de preferencias).
2. Se nos abrirá en una carpeta “MiniZincIDE”. Iremos a la anterior “SoftPR”, y de allí a “MiniZinc” y “bin”. Pulsamos “seleccionar carpeta”
3. Ahora ya podemos dar “Close” y empezar a trabajar.

1) [4] Un grafo se dice bipartito si sus vértices se pueden pintar en dos colores, de forma que cada arista una vértices en colores distintos:



Queremos escribir un modelo MiniZinc que nos diga si el siguiente grafo es bipartito:



Para ello:

- a) Representaremos los vértices A,B,D,E,F,G,H como variables MiniZinc de tipo bool. Es decir en nuestro modelo true representa un color y false el otro.
- b) Escribiremos 12 restricciones (una por arista) que indique que las variables que une la arista deben ser

diferentes (representando diferente color). Se pueden escribir mediante conjunciones (es decir no hace falta escribir 12 instrucciones constraint, se pueden unir varias con el operador \wedge)

Incluir en la solución el modelo generado.

2) [4] Queremos pintar 10 coches en secuencia cada uno tiene un id. marcado con un número en 1 y 10. Para determinar las posiciones se almacenará en un array de 10 enteros los valores de la secuencia 9 8 7 6 5 4 3 2 1 es una posible solución, que indica que el coche 1 se hará el noveno, el coche 2 el octavo en la secuencia, y así hasta el coche marcado con 10, que se hará el primero. Es decir el $a[i] = j$ indica que el coche marcado con i se pintará en la posición j . Escribir un programa con restricciones: Se debe cumplir:

- a) Todos los coches aparecerán en la secuencia una y solo una vez.
- b) Los coches con marcas 3,4,5 se deben pintar consecutivos y en este orden.
- c) El coche 10 se debe pintar el primero
- d) El coche 6 no puede ser el último.
- e) El coche 7 tiene que pintarse después del 6 ,5 y 8 (no inmediatamente ni a continuación, solo “detrás” en algún momento).
- f) Los coches 8 y 9 no pueden pintarse consecutivos.

Escribe el modelo en la solución

3) [1] Añadir al programa anterior la siguiente restricción:

constraint $a[3] = 9$

¿Hay soluciones al nuevo modelo? ¿Qué significa? (una frase debe bastar).

4) [1] Dos vectores del mismo tamaño se dicen “rotaciones” si los valores de uno se obtienen desplazando circularmente los del otro. Por ejemplo 4 6 8 9 1 y 9 1 4 6 8 son rotaciones. Escribir un modelo que a partir de un vector de parámetros (constante) obtenga todos los vectores rotación posibles.

Añadir el modelo a la solución.