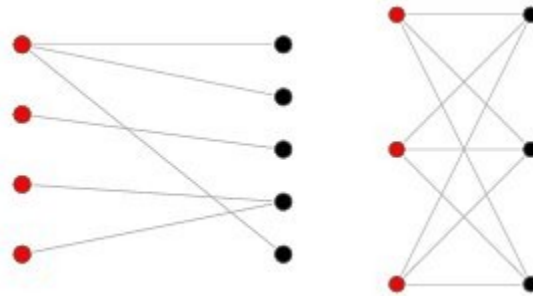
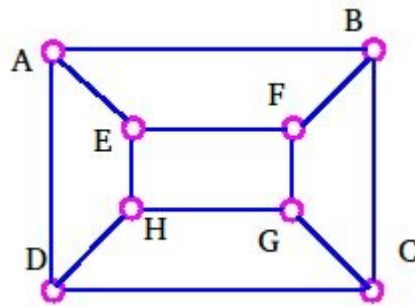


Práctica 1 – Generación Automática de Casos de Prueba

4) [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:

- 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.
- 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 generar casos de prueba para el siguiente programa:

```
int maximo(int x, int y, int z) {
    m = 0;
    if (x<y) { // máximo y o z
        if (y<z)
            m = z;
        else
            m = y
    } else // máximo x o z
        if (x<z)
            m=z
        else
            m=x;

    return m;
}
```

Para ello queremos escribir un modelo MiniZinc que modele las relaciones entre las variables.

a) Las variables serán x, y, z, m . Las declaramos en el rango $-2..2$.

b) Escribimos una restricción que muestre la relación que cumplen las 4 variables. Para ello escribiremos una constraint que defina la disyunción de las 4 posibilidades. Como pista incluimos el último ejemplo:

constraint (...) \vee (...) \vee (...) \vee (not($x < y$) \wedge not($x < z$) \wedge $m = x$);

Escribir en la solución la restricción completa.

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

constraint $m \neq x \wedge m \neq y \wedge m \neq z$;

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

4) [1] Ahora añadimos una nueva función al código.

```
int f(int a, int b) {
    int u;
    if (a > 0)
        u = maximo(a, b, 100);
    else
        u = maximo(a*a, b, b*b)+1;

    return u;
}
```

Añadir al modelo:

a) Las variables a, b, u de tipo entero.

b) Una restricción que modele los posibles valores de f (tener en cuenta el paso de parámetros)

Pista: el nuevo modelo debe tener exactamente 6 soluciones

Incluir en la solución la nueva restricción.