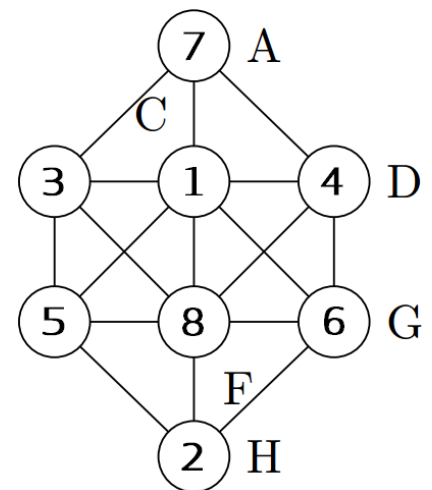


Aufgabe 1

- a) **gesucht:** geeignete CSP-Modellierung zur Lösung des abgebildeten Graphen, in dem Nachbarn keine aufeinanderfolgenden Zahlen beinhalten dürfen

Lösung: Wir gehen nach dem Prinzip „Most-constrained (MC)“ vor, so dass wir bei den Knoten mit höchstem Grad beginnen, hier den beiden mittleren.

- Die Knoten C und F haben je sechs Nachbar. Würden wir ihnen eine der Zahlen 2 bis 7 zuordnen, blieben für die sechs Nachbarn nur noch fünf Zahlen übrig (die zugeordnete sowie die beiden benachbarten Zahlen wären gesperrt). Dementsprechend erhalten C und F die Zahlen 1 und 8 (zwei Möglichkeiten). Wir setzen **C = 1** und **F = 8**.
- Die Knoten B, D, E und G haben je vier Nachbarn. Die Zahlen 2 und 7 scheiden an jedem Knoten aus, also bleiben für die vier Knoten nur noch die Zahlen 3, 4, 5 und 6 übrig. Für die Knoten A und H bleiben damit die Zahlen 2 und 7. Auf Grund der Nachbarschaftsbeziehungen müssen wir **A = 7** und **H = 2** setzen.
- Wegen A = 7 müssen B und D aus {3,4,5} und wegen H = 2 müssen E und G aus {4,5,6} gewählt werden. B und E (sowie D und G) sind benachbart. Wäre B = 5, bliebe für E nichts mehr übrig (analog D = 5 und G). Wäre umgekehrt E = 4, bliebe für B nichts mehr übrig (analog G = 4 und D). Damit können wir die Domänen einschränken: B und D aus {3,4}, E und G aus {5,6}. Die „Besitzer“ von 4 und 5 dürfen nicht benachbart sein, so dass eine Lösung durch **B = 3, D = 4, E = 5** und **G = 6** gegeben ist (die einzige andere durch 4, 3, 6, 5 in der Reihenfolge).
- Damit haben wir zweimal zwei Variationsmöglichkeiten, die horizontaler und vertikaler Spiegelung entsprechen – insgesamt also **vier Lösungen**.



- b) Folgender Code ermittelt in MiniZinc die vertikal und horizontal gespiegelte Lösung A=2, B=6, C=8, D=5, E=4, F=1, G=3, H=7:

```
include "alldifferent.mzn";
int: nc = 8;

var 1..nc: A; var 1..nc: B; var 1..nc: C; var 1..nc: D;
var 1..nc: E; var 1..nc: F; var 1..nc: G; var 1..nc: H;

constraint A != B+1; constraint A+1 != B;
constraint A != C+1; constraint A+1 != C;
constraint A != D+1; constraint A+1 != D;
constraint B != C+1; constraint B+1 != C;
constraint B != E+1; constraint B+1 != E;
```

```

constraint B != F+1; constraint B+1 != F;
constraint C != E+1; constraint C+1 != E;
constraint C != F+1; constraint C+1 != F;
constraint C != G+1; constraint C+1 != G;
constraint D != C+1; constraint D+1 != C;
constraint D != F+1; constraint D+1 != F;
constraint D != G+1; constraint D+1 != G;
constraint E != F+1; constraint E+1 != F;
constraint E != H+1; constraint E+1 != H;
constraint F != G+1; constraint F+1 != G;
constraint F != H+1; constraint F+1 != H;
constraint G != H+1; constraint G+1 != H;

constraint alldifferent([A,B,C,D,E,F,G,H]);
solve satisfy;

output [" A=", show(A), " B=", show(B), " C=", show(C), " D=", show(D),
        " E=", show(E), " F=", show(F), " G=", show(G), " H=", show(H)]

```

c) **zu zeigen:** alle Constraints lassen sich auf binäre Constraints darstellen

Beweis: Wir betrachten zunächst beispielhaft die Restriktion $a + b = c$. Wir führen eine neue Variable „ab“ ein, die die Form eines zweielementigen Vektors hat. Mit den beiden Restriktionen **elem(a,ab,1)** und **elem(b,ab,2)** können wir sicherstellen, dass die Variable ab aus den Werten für a und b besteht. Nun müssen wir uns nur noch eine Restriktion definieren, die die obige Gleichung abbildet. In diesem Fall sei **sum(ab,c)** der Wahrheitswert von „c ist die Summe der Elemente in ab“.

Mit diesem Verfahren lässt sich jede (n+1)-äre Restriktion auf eine n-äre Restriktion reduzieren: Zunächst werden je zwei Elemente mit „elem“ in einer Vektorvariablen gespeichert. Mit einer dritten Restriktion wird die entsprechende Verknüpfung abgebildet (sum, diff, prod, quot etc.). Per Induktion ergibt sich so nach n-1 solchen Umformungen schließlich eine binäre Restriktion. \square

d) **zu zeigen:** Die Zuordnung **WA = rot und V = blau** lässt sich zu keiner gültigen Färbung der australischen Bundesstaaten in drei Farben erweitern

Beweis: Da wir Tasmanien beliebig färben können, lassen wir es in der Tabelle ohne Beschränkung der Allgemeinheit weg. In den neu hinzugekommenen roten Feldern ist jeweils angegeben, warum das jeweilige Feld rot ist (entweder weil für den Staat bereits die angegebene Farbe zugeordnet wurde oder weil sich durch die Nachbarschaft mit dem angegebenen Staat die entsprechende Unmöglichkeit der Zuordnung ergibt).

	rot	blau	grün
WA		(rot)	(rot)
NT	(WA)		
SA	(WA)	(V)	
Q			
NSW		(V)	
V	(blau)		(blau)

	rot	blau	grün
WA			
NT			(SA)
SA			
Q			(SA)
NSW			(SA)
V			

	rot	blau	grün
WA			
NT			
SA			
Q	(NSW)	(NT)	
NSW			
V			

Somit bleibt für Queensland keine mögliche Zuordnung mehr übrig, so dass mit gegebener Teilzuordnung keine Färbung in drei Farben möglich ist. □