Artificial Intelligence

CSC 665

tyler dae devlin

CSPs I

9.21.2023

Map coloring

This map uses seven colors. Can we do it in fewer?

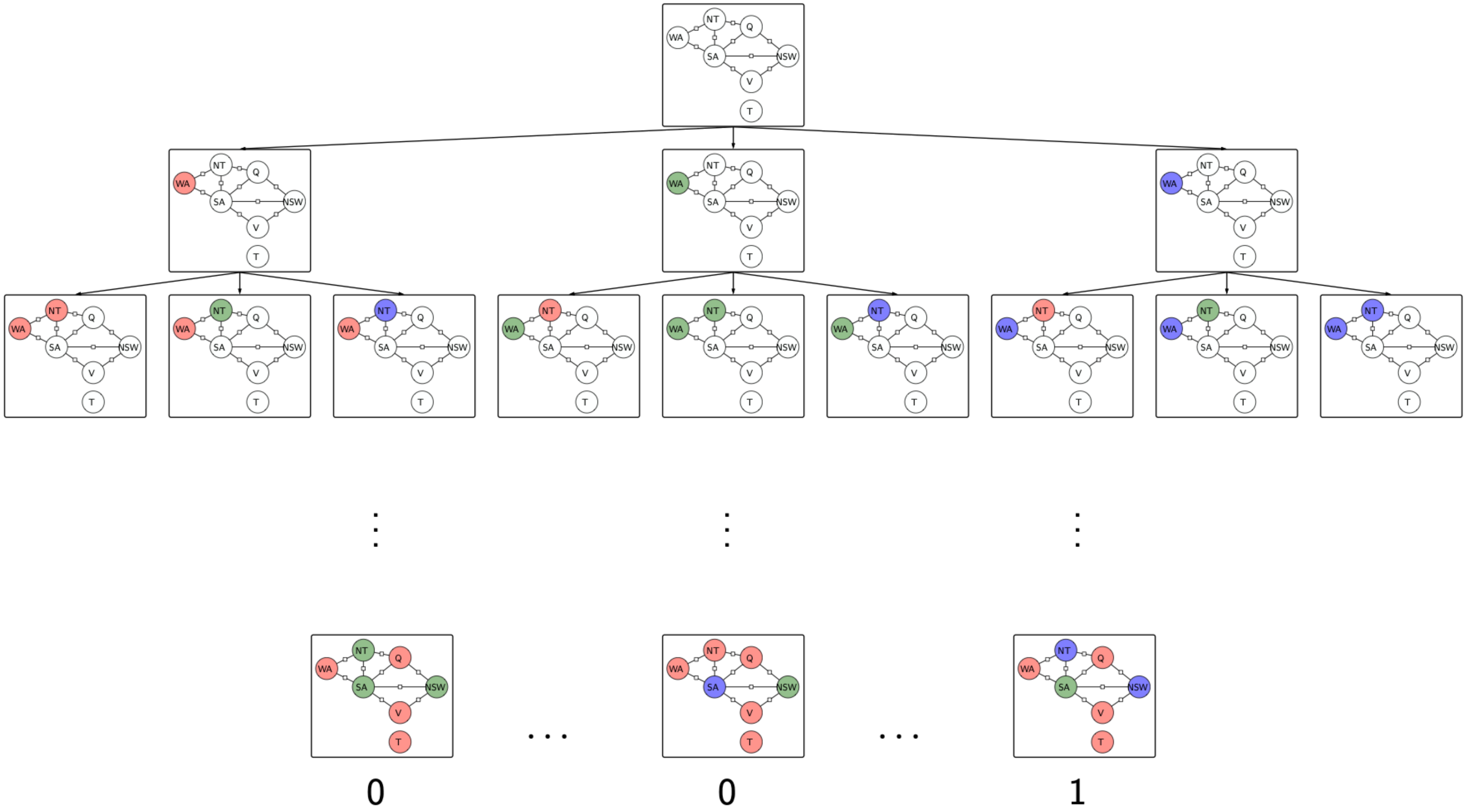

NORTHERN TERRITORY
QUEENSLAND
WESTERN AUSTRALIA
SOUTH AUSTRALIA
NEW SOUTH WALES
VICTORIA
TASMANIA

Yes!

- **Search:** make decisions by looking ahead
- **Logic:** deduce new facts from existing facts
- **Constraints:** find a way to satisfy a given specification
- **Probability:** reason quantitatively about uncertainty
- **Learning:** make future predictions from past observations

Modeling map coloring

- How can we solve this problem algorithmically?
- Logic?
- Search?
  - **States** are partially colored maps
  - **Start state** is the empty map
  - **Action:** assign the next region a color
  - **End state:** all regions colored with no neighboring conflicts


WA
NT
Q
SA
NSW
V
T
0
0
1

Modeling map coloring

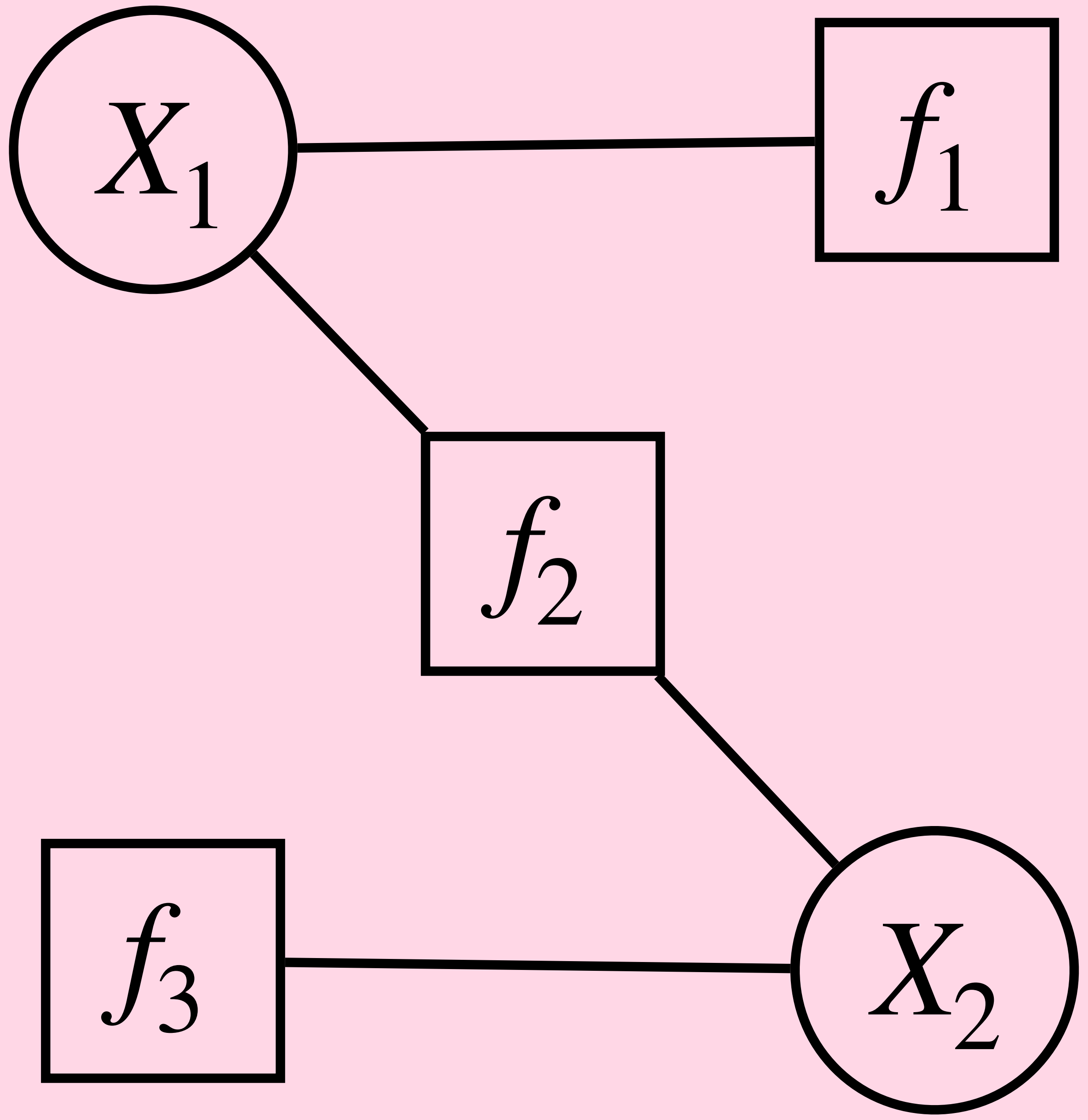
- Treating this as a search problem is equivalent to enumerating all possible colorings
- Can we do better?
- One approach: put more domain information into the **model**
  - Choose a better ordering of regions
  - Restrict actions to valid coloring assignments given current state
- Alternative approach: keep the model simple and push more work onto the **inference** algorithms

Modeling: factor graphs

New representation: factor graphs

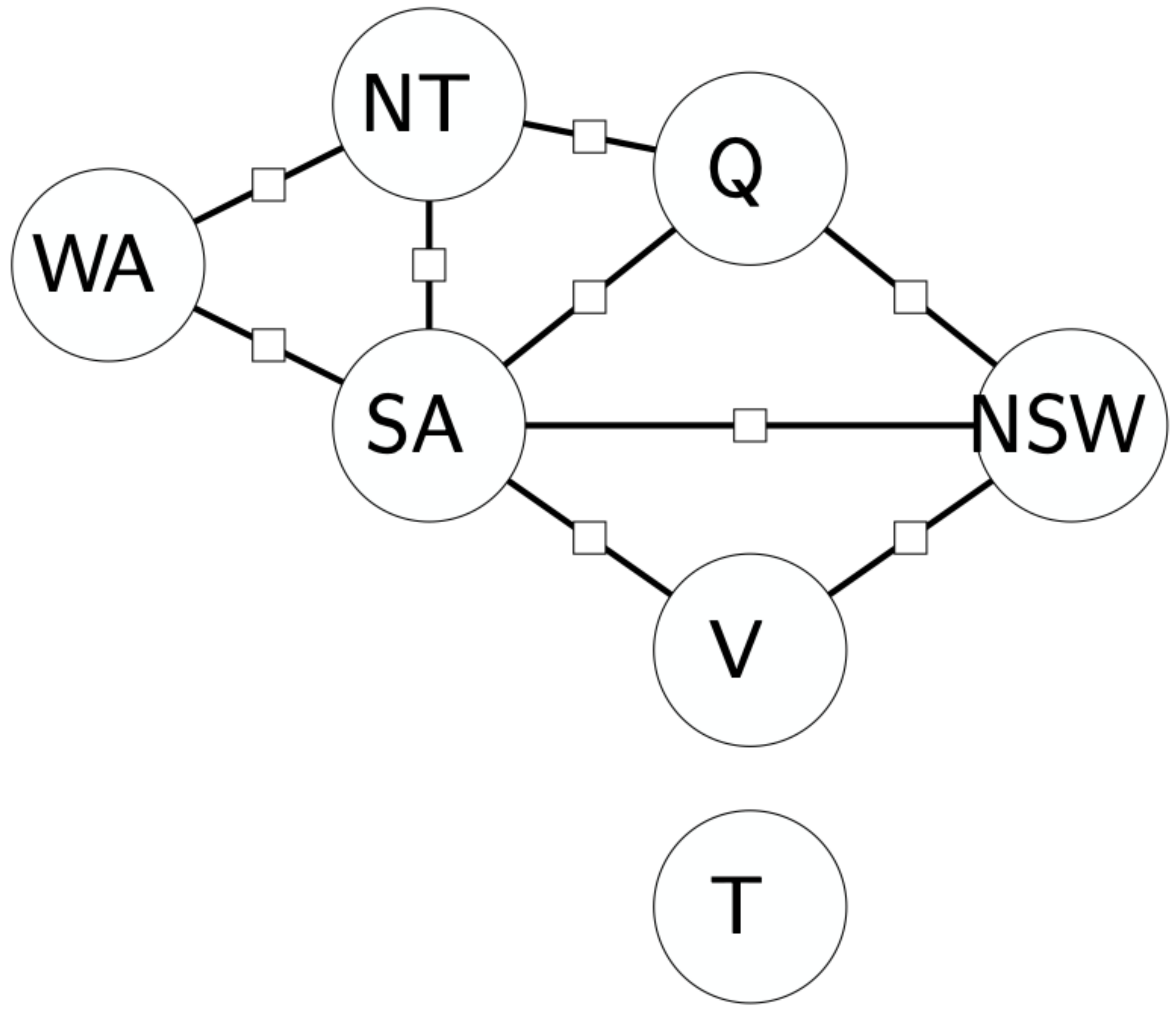
A **factor graph** consists of

- Variables $X = (X_1, X_2, \ldots, X_n)$
- Domains $D = (D_1, D_2, \ldots, D_n)$, where $X_i \in D_i$
- Factors $f_1, f_2, \ldots, f_m$ where $f_j(X) \geq 0$

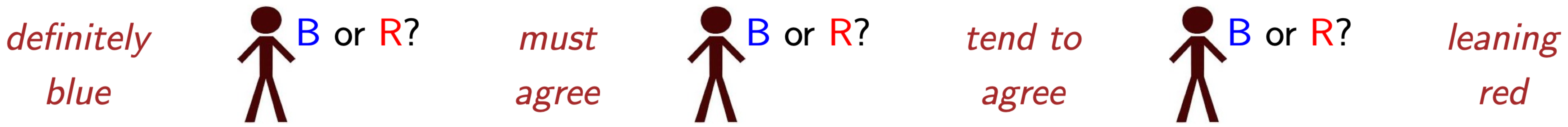


Example: Australia

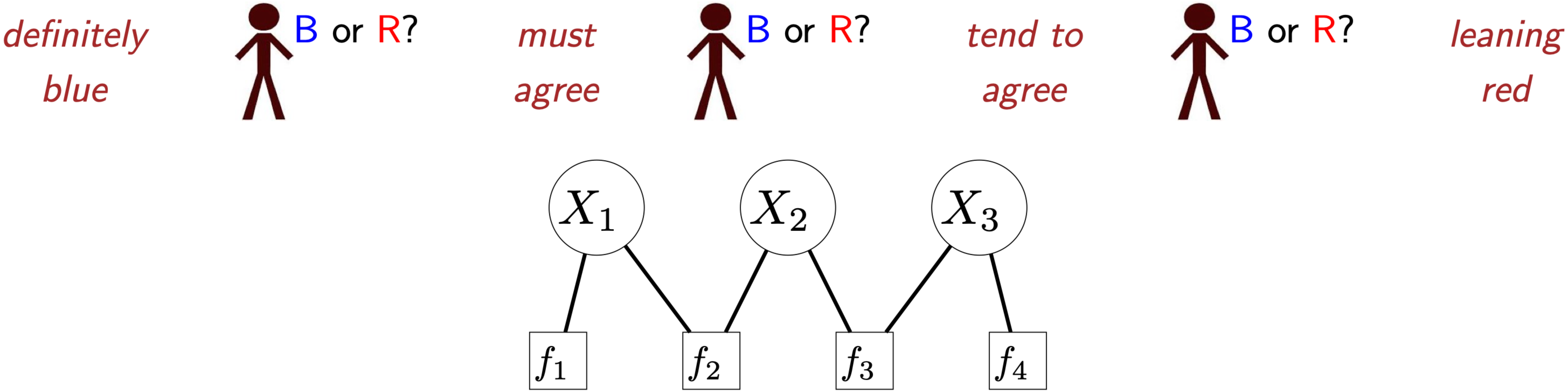
- $X = (\text{WA}, \text{NT}, \text{SA}, \text{Q}, \text{NSW}, \text{V}, \text{T})$
- $D_i = \{R, G, B\}$
- Factors
  - $f_1(X) = \mathbf{1}\{\text{WA} \neq \text{NT}\}$
  - $f_2(X) = \mathbf{1}\{\text{NT} \neq \text{Q}\}$
  - ...



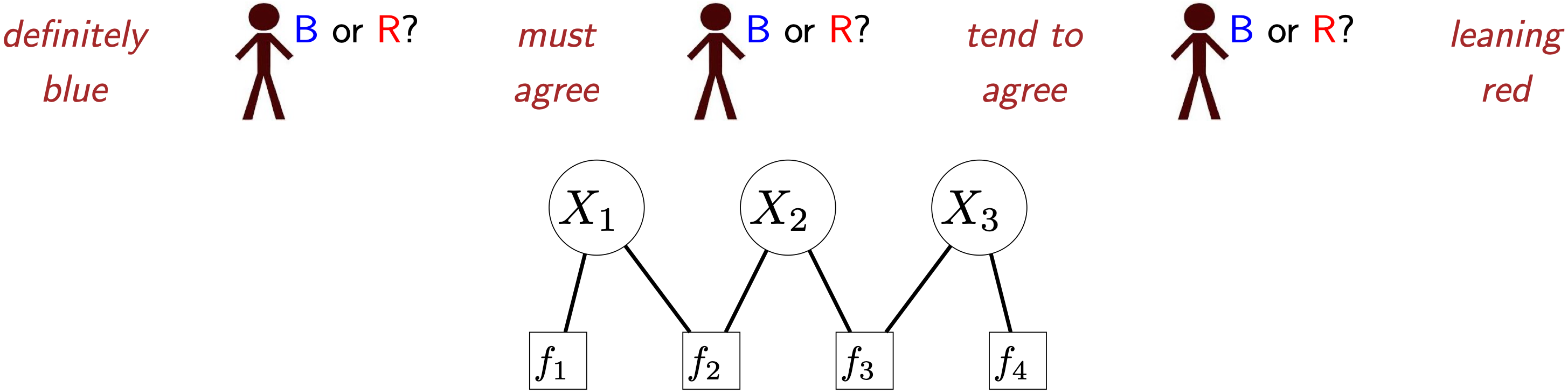
Example: voting



Example: voting



Example: voting



x_1	$f_1(x_1)$
R	0
B	1

x_1	x_2	$f_2(x_1, x_2)$
R	R	1
R	B	0
B	R	0
B	B	1

x_2	x_3	$f_3(x_2, x_3)$
R	R	3
R	B	2
B	R	2
B	B	3

x_3	$f_4(x_3)$
R	2
B	1

The factor graph tells us how to score ("weight") joint assignments of variables to values in their domains.

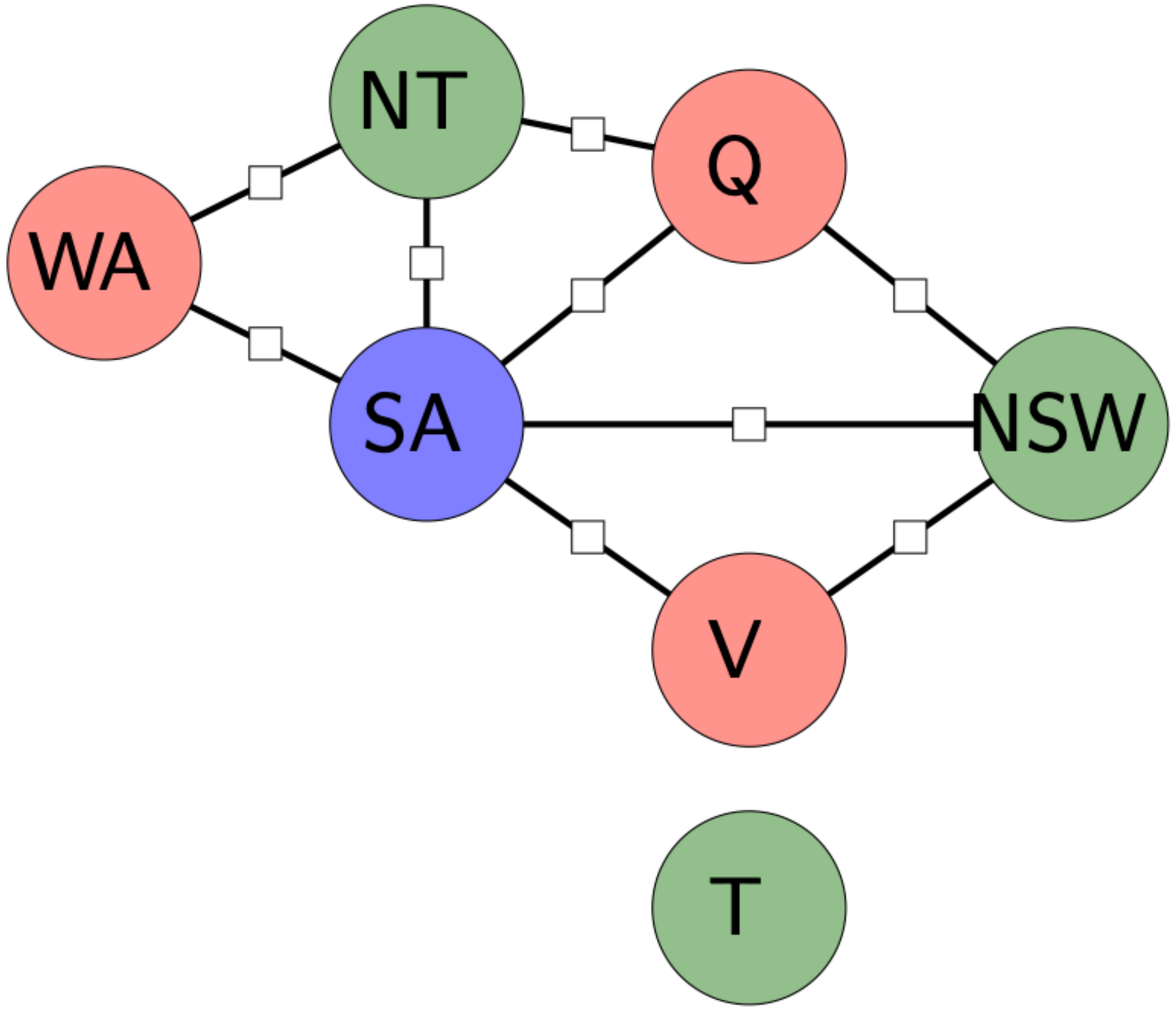
x_1	$f_1(x_1)$
R	0
B	1

x_1	x_2	$f_2(x_1, x_2)$
R	R	1
R	B	0
B	R	0
B	B	1

x_2	x_3	$f_3(x_2, x_3)$
R	R	3
R	B	2
B	R	2
B	B	3

x_3	$f_4(x_3)$
R	2
B	1

x_1	x_2	x_3	Weight
R	R	R	$0 \cdot 1 \cdot 3 \cdot 2 = 0$
R	R	B	$0 \cdot 1 \cdot 2 \cdot 1 = 0$
R	B	R	$0 \cdot 0 \cdot 2 \cdot 2 = 0$
R	B	B	$0 \cdot 0 \cdot 3 \cdot 1 = 0$
B	R	R	$1 \cdot 0 \cdot 3 \cdot 2 = 0$
B	R	B	$1 \cdot 0 \cdot 2 \cdot 1 = 0$
B	B	R	$1 \cdot 1 \cdot 2 \cdot 2 = 4$
B	B	B	$1 \cdot 1 \cdot 3 \cdot 1 = 3$



Only valid colorings will have nonzero weight.

All colorings that violate the neighbor constraint will have at least one zero factor in the product.

Assignment weight

The assignment of values $x = (x_1, \ldots, x_n)$ to variables $X = (X_1, \ldots, X_n)$ has **weight**

$$W(x) = f_1(x) f_2(x) \ldots f_m(x)$$
$$= \prod_{j=1}^{m} f_j(x)$$

As assignment x is **consistent** if $W(x) > 0$

If multiple assignments are consistent, want to find the **highest weight** assignment

A CSP is **satisfiable** if there is some consistent assignment