# 人工智能基础 HW3

PB18111697 王章瀚

2021 年 4 月 14 日

## 6.5

**Solve the cryptarithmetic problem in Figure 6.2 by hand, using the strategy of backtracking with forward checking and the MRV and least-constraining-value heuristics respectively.**

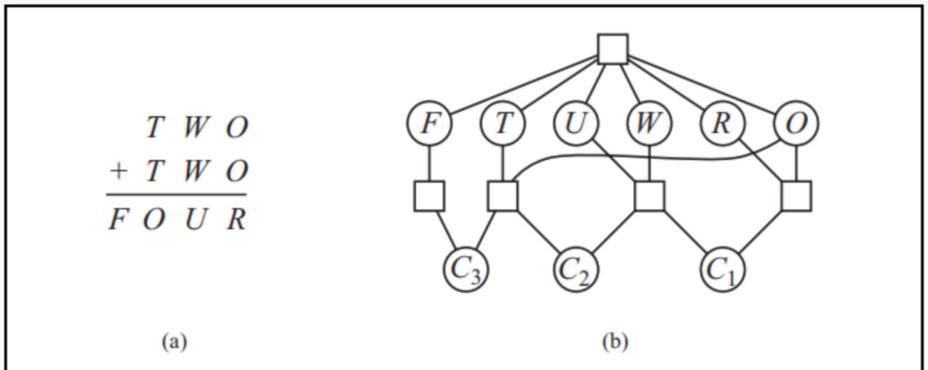


**Figure 6.2** (a) A cryptarithmetic problem. Each letter stands for a distinct digit; the aim is to find a substitution of digits for letters such that the resulting sum is arithmetically correct, with the added restriction that no leading zeroes are allowed. (b) The constraint hypergraph for the cryptarithmetic problem, showing the *Alldiff* constraint (square box at the top) as well as the column addition constraints (four square boxes in the middle). The variables $C_1$, $C_2$, and $C_3$ represent the carry digits for the three columns.

First of all, the **domains** are

- $C_1$: $\{1\}$
- $C_2, C_3$: $\{0, 1\}$
- others: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

And the **constraints** are as following:

- $F, T, U, W, R, O$ are different
- $C_1 = 2 \times O//10$, $R = 2 \times O\%10$, here "//" means an integer division with remainder, thus $C_1$ is an interger quotient.
- $C_2 = (2 \times W + C_1)//10$, $U = (2 \times W + C_1)\%10$
- $C_3 = (2 \times T + C_2)//10$, $O = (2 \times T + C_2)\%10$

- $F = C_3$

**The 3 methods (forward checking, MRV, least-constraining-value(LCV)) are compatible, so I'll use them all.**

1. choose $C_3$ (MRV); choose 1 for $C_3$ (LCV);
   domain of $F$ becomes $\{1\}$,
   domain of $T$ becomes $\{5, 6, 7, 8, 9\}$

2. choose $F$ (MRV); choose 1 for $F$ (only choice);
   remove 1 from domain of others due to alldiff

3. choose $C_2$ (MRV); choose 0 for $C_2$ (LCV);
   domain of $W$ becomes $\{0, 2, 3, 4\}$,
   domain of $O$ becomes $\{0, 4, 6, 8\}$ (must be even)

4. choose $C_1$ (MRV); choose 0 for $C_1$ (LCV);
   domain of $O$ becomes $\{0, 4\}$,
   domain of $R$ becomes $\{0, 8\}$,
   domain of $U$ becomes $\{0, 4, 6, 8\}$

5. choose $O$ (MRV); choose 4 for $O$ (LCV);
   domain of $R$ becomes $\{8\}$;
   domain of $T$ becomes $\{7\}$ remove 4 from domain of others due to alldiff

6. choose $R$ (MRV); choose 8 for $R$ (only choice);
   remove 8 from domain of others due to alldiff

7. choose $T$ (MRV); choose 7 for $T$ (only choice);
   remove 7 from domain of others due to alldiff

8. choose $U$ (MRV); choose 6 for $U$ (forward checking). Otherwise, $U$ has to be 0, causing $W$ being 0, which is not allowed.
   domain of $W$ becomes $\{3\}$

9. choose $W$ (MRV); choose 3 for $W$ (only choice)

After steps above, we find a **solution**:

$$F = 1, T = 7, O = 4, U = 6, W = 3, R = 8$$

## 6.11

**Use the AC-3 algorithm to show that arc consistency can detect the inconsistency of partial assignment $WA = green$, $V = red$ for the problem shown in Figure 6.1.**

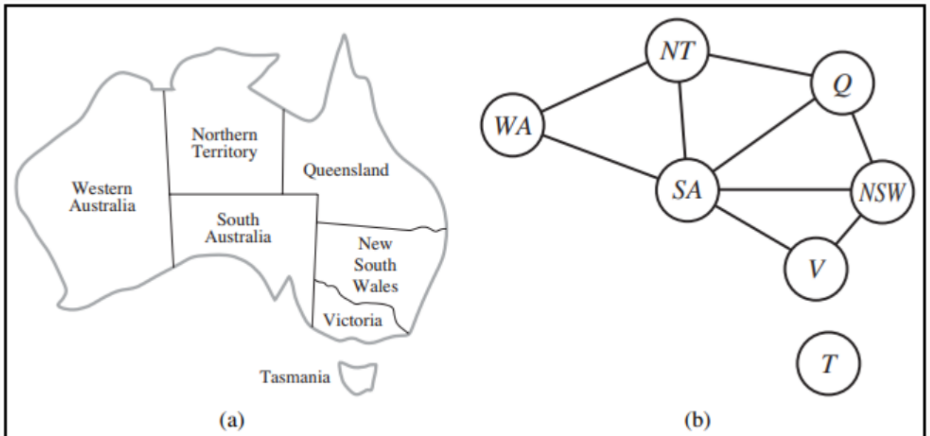


**Figure 6.1** (a) The principal states and territories of Australia. Coloring this map can be viewed as a constraint satisfaction problem (CSP). The goal is to assign colors to each region so that no neighboring regions have the same color. (b) The map-coloring problem represented as a constraint graph.

The steps of running AC-3 algorithm is as following:

1. the queue contains all arcs:

$$\{(WA, NT), (WA, SA), (NT, Q), (NT, SA), (NT, WA),$$
$$(SA, WA), (SA, NT), (SA, Q), (SA, NSW), (SA, V),$$
$$(Q, NT), (Q, SA), (Q, NSW),$$
$$(NSW, Q), (NSW, SA), (NSW, V), (V, SA), (V, NSW)\}$$

2. after checking all elements above, the remaining legal assignments are:

| WA | NT | SA | Q | NSW | V | T |
|---|---|---|---|---|---|---|
| $g$ | $r\ b$ | $b$ | $rgb$ | $gb$ | $r$ | $rgb$ |

   and then we still need to check:

$$\{(NT, Q), (NT, SA), (NT, WA)$$
$$(SA, WA), (SA, NT), (SA, Q), (SA, NSW), (SA, V),$$
$$(NSW, Q), (NSW, SA), (NSW, V)\}$$

3. after checking all elements above, the remaining legal assignments are:

| WA | NT | SA | Q | NSW | V | T |
|---|---|---|---|---|---|---|
| $g$ | $r\ b$ | $b$ | $r$ | $g$ | $r$ | $rgb$ |

   and then we still need to check:

$$\{(Q, NT), (Q, SA), (Q, NSW),$$
$$(NSW, Q), (NSW, SA), (NSW, V)\}$$

4. after checking all elements above, the remaining legal assignments are:

| WA | NT | SA | Q | NSW | V | T |
|---|---|---|---|---|---|---|
| $g$ | $b$ | $b$ | $r$ | $g$ | $r$ | $rgb$ |

   and then we still need to check:

$$\{(NT, Q), (NT, SA), (NT, WA)\}$$

5. this time, $(NT, SA)$ will be inconsistent(both have the only choice of blue), thus we know AC-3 algorithm can detect the inconsitency of this circumstance.

## 6.12

**What is the worst-case complexity of running AC-3 on a tree-structured CSP?**

Suppose that there are $E$ edges in this tree-structured CSP.

If nodes' domains are changed(can only be decreased) in each iteration of a whole queue(excluding newly added ones), then we need check $E$ more arcs per-iteration.

Thus, suppose $D$ is the largest domain size, then the worst-case complexity of running AC-3 on a tree-structured CSP is $O(ED)$