

# Discrete Optimization

Constraint Programming: Part II

# Goals of the Lecture

- ▶ Illustrating more complex constraint propagation
- ▶ Showing that constraints have dedicated algorithms

# Computational Paradigm

- Branch and prune
  - pruning
    - reduce the search space as much as possible
  - branching
    - decompose the problem into subproblems and explore the subproblems

# Computational Paradigm

- ▶ Branch and prune
  - pruning
    - reduce the search space as much as possible
  - branching
    - decompose the problem into subproblems and explore the subproblems
- ▶ Pruning
  - use constraints to remove, from the variable domains, values that cannot belong to any solution

# Computational Paradigm

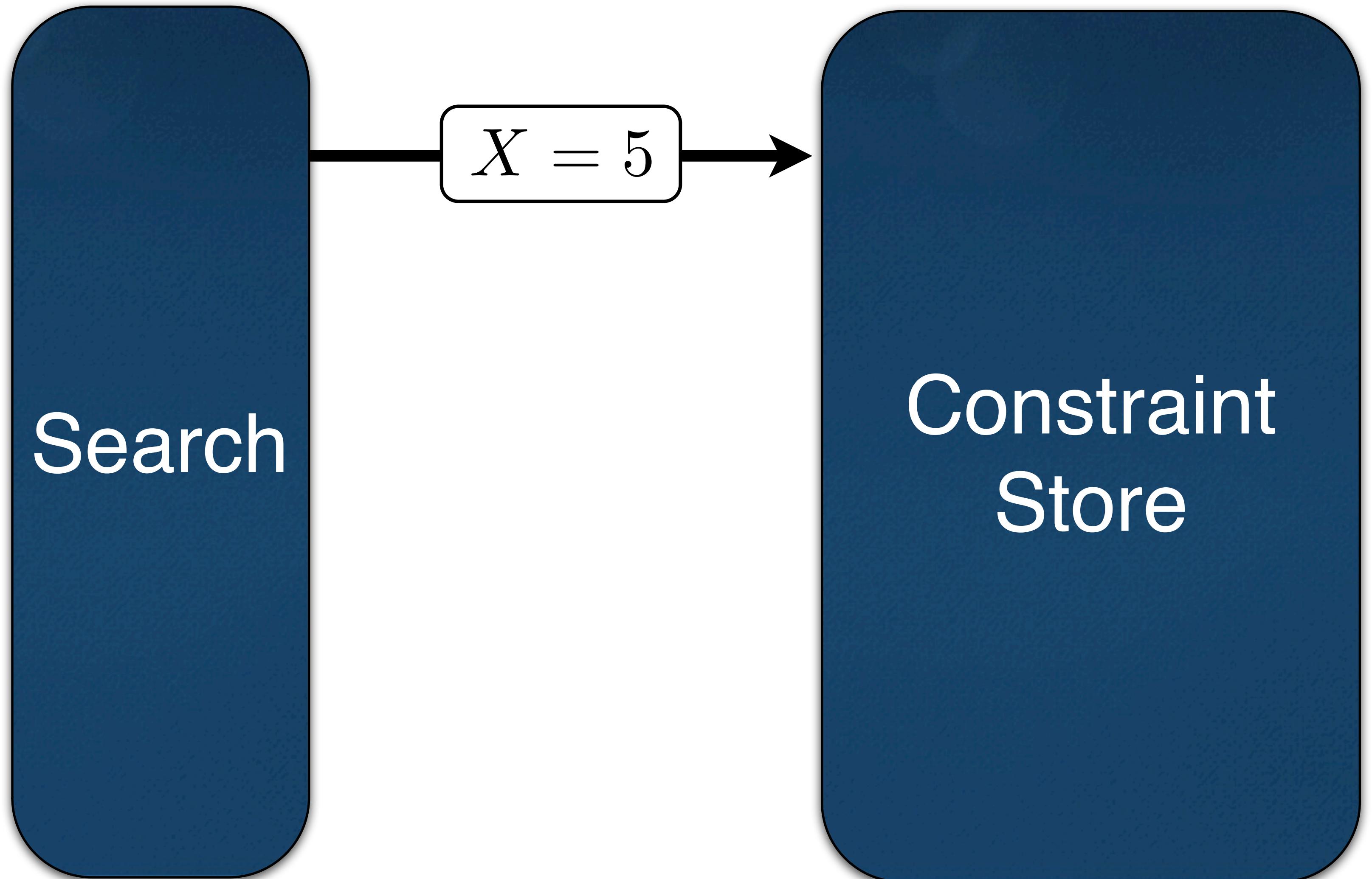
- ▶ Branch and prune
  - pruning
    - reduce the search space as much as possible
  - branching
    - decompose the problem into subproblems and explore the subproblems
- ▶ Pruning
  - use constraints to remove, from the variable domains, values that cannot belong to any solution
- ▶ Branching
  - e.g., try all the possible values of a variable until a solution is found or it can be proven that no solution exists

# Computational Paradigm

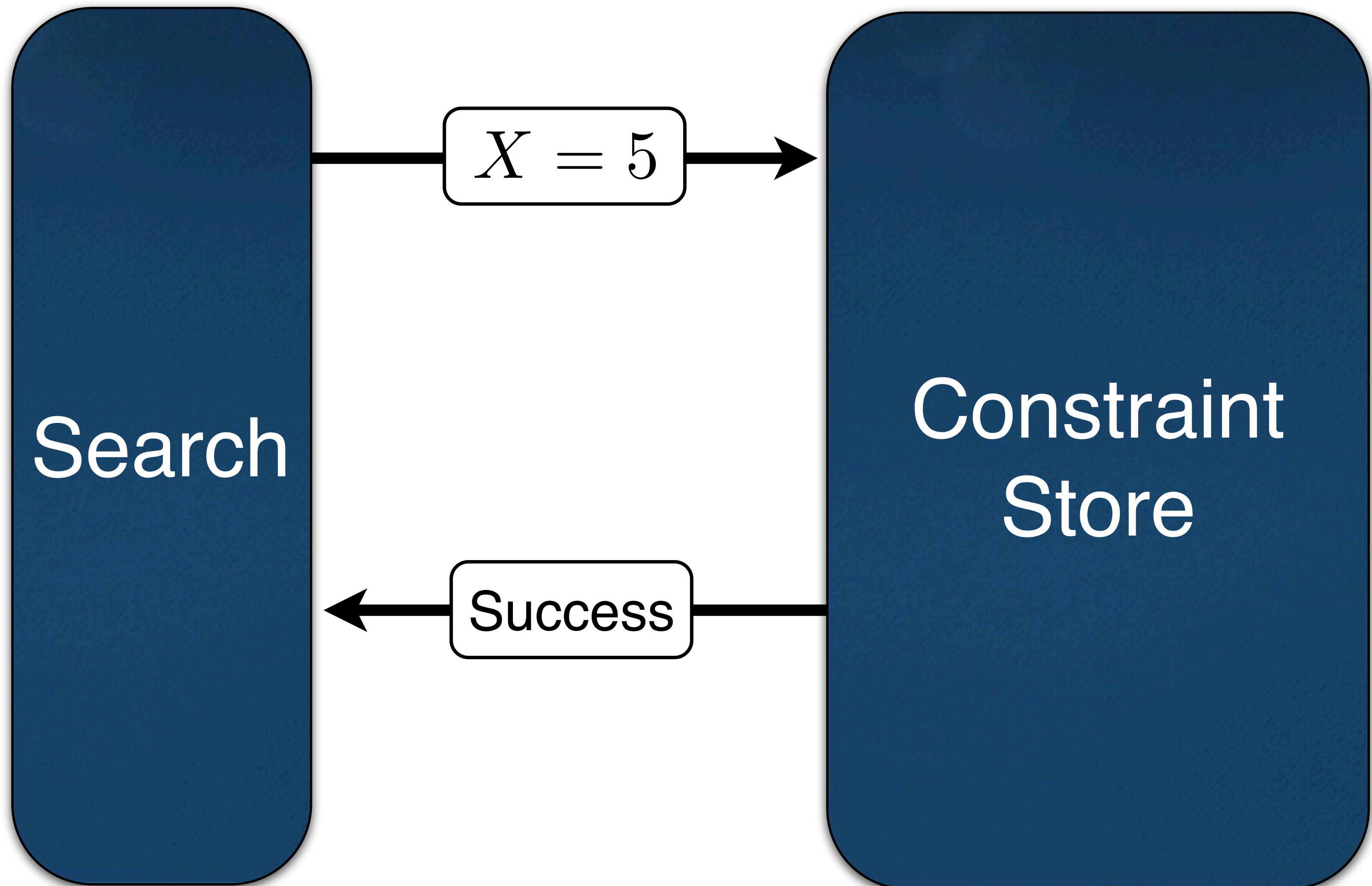
Search

Constraint  
Store

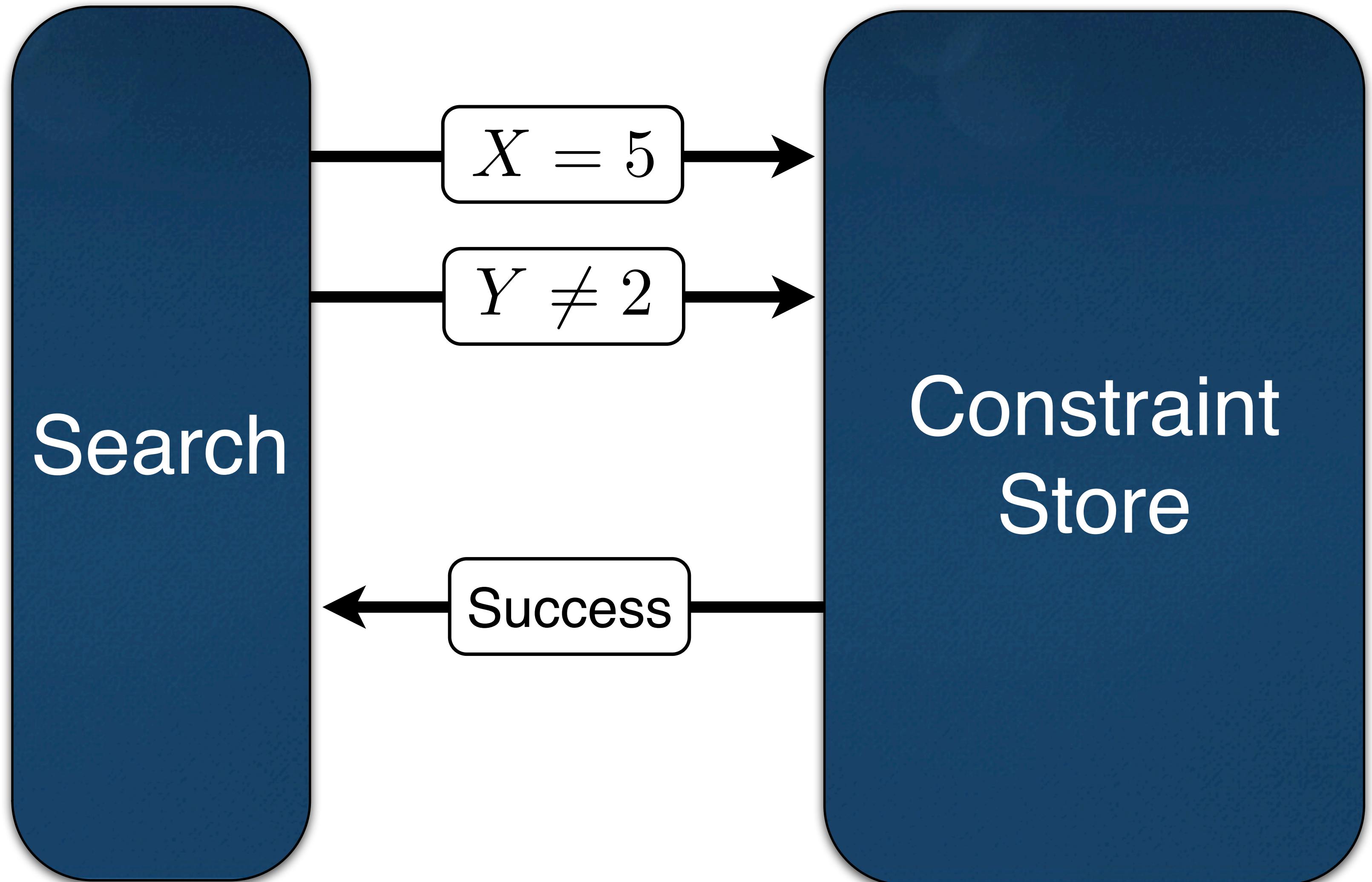
# Computational Paradigm



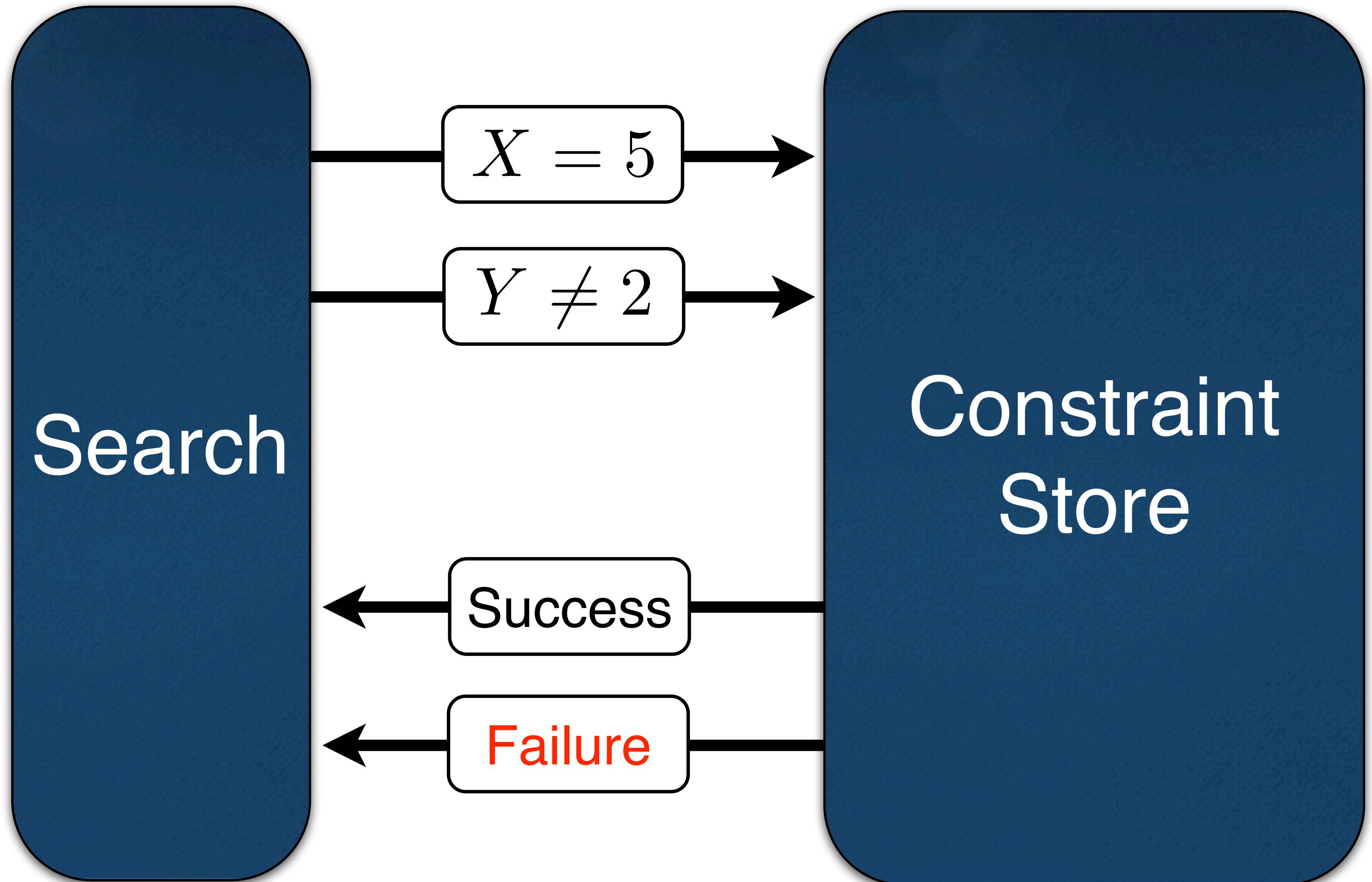
# Computational Paradigm



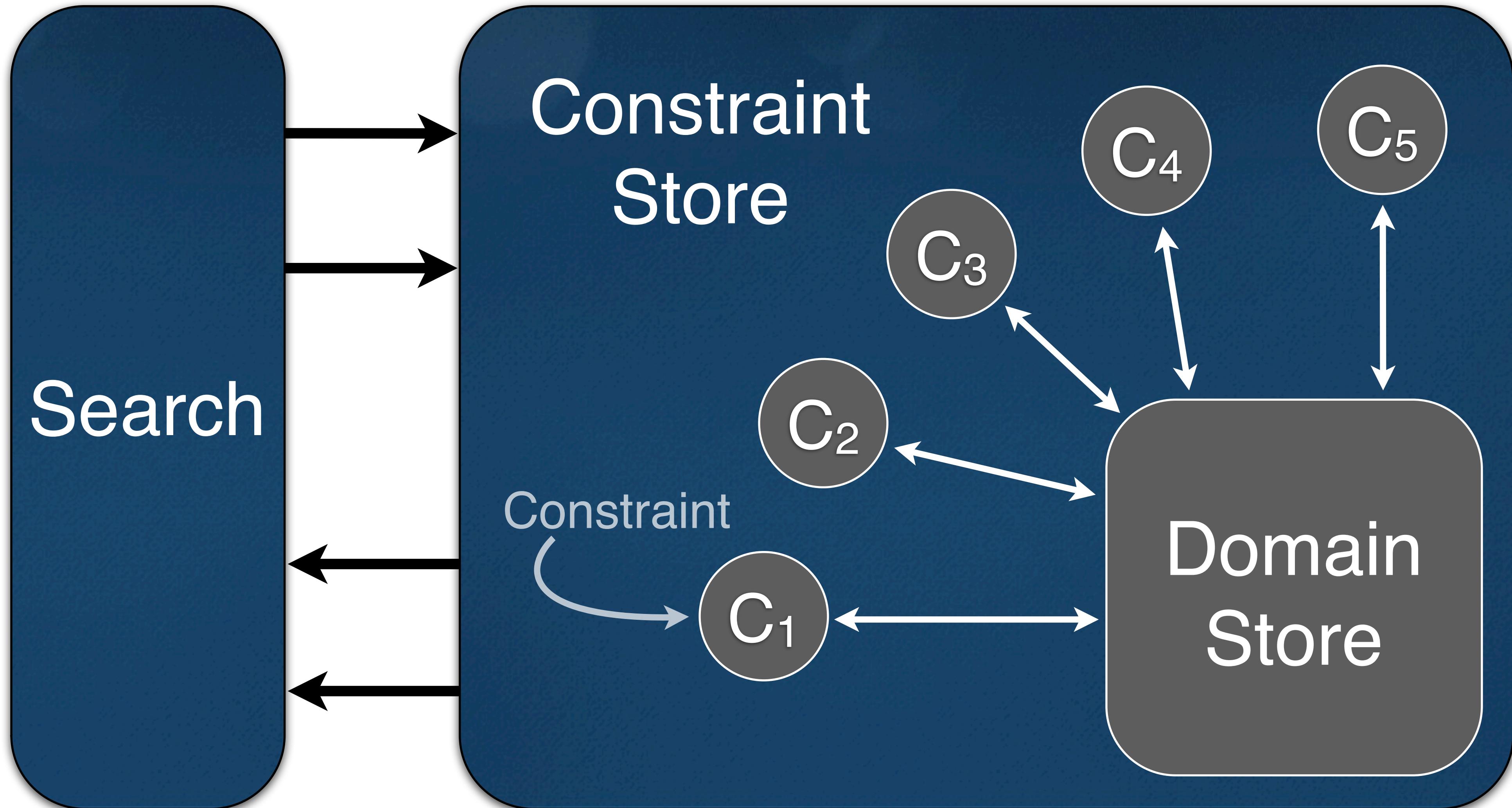
# Computational Paradigm



# Computational Paradigm



# Computational Paradigm



# Computational Paradigm

## ► What does a constraint do?

- feasibility checking
- pruning

# Computational Paradigm

- ▶ What does a constraint do?

- feasibility checking
- pruning

- ▶ Feasibility checking

- can a constraint be satisfied given the values in the domains of its variables

# Computational Paradigm

- ▶ What does a constraint do?

- feasibility checking
- pruning

- ▶ Feasibility checking

- can a constraint be satisfied given the values in the domains of its variables

- ▶ Pruning

- if satisfiable, determine which values in the domains cannot be part of any solution

# Computational Paradigm

- ▶ What does a constraint do?
  - feasibility checking
  - pruning
- ▶ Feasibility checking
  - can a constraint be satisfied given the values in the domains of its variables
- ▶ Pruning
  - if satisfiable, determine which values in the domains cannot be part of any solution
- ▶ The algorithms use dedicated algorithms for each constraint
  - they exploit the structure and properties of the constraint

# Send More Money

## ► Specification

- assign different digits to letters to satisfy the addition

$$\begin{array}{r} \text{S E N D} \\ + \text{M O R E} \\ \hline = \text{M O N E Y} \end{array}$$

# Send More Money

- ▶ Specification
  - assign different digits to letters to satisfy the addition
- ▶ Model to illustrate constraint propagation
  - no claim that this is a good model: it is not

$$\begin{array}{r} \text{S E N D} \\ + \text{M O R E} \\ \hline = \text{M O N E Y} \end{array}$$

# Send More Money

- Basic modeling
  - add carries explicitly like in kindergarten

$$\begin{array}{r} \text{C}_4 \quad \text{C}_3 \quad \text{C}_2 \quad \text{C}_1 \\ \text{S} \quad \text{E} \quad \text{N} \quad \text{D} \\ + \quad \text{M} \quad \text{O} \quad \text{R} \quad \text{E} \\ \hline \\ = \quad \text{M} \quad \text{O} \quad \text{N} \quad \text{E} \quad \text{Y} \end{array}$$

# Send More Money

- ▶ Basic modeling
  - add carries explicitly like in kindergarten
- ▶ What are the decision variables
  - there is a variable for each letter to denote the value of the letters
  - there is a variable for each carry

$$\begin{array}{r} C_4 \quad C_3 \quad C_2 \quad C_1 \\ S \quad E \quad N \quad D \\ + \quad M \quad O \quad R \quad E \\ \hline \\ = \quad M \quad O \quad N \quad E \quad Y \end{array}$$

# Send More Money

```
enum Letters = { S, E, N, D, M, O, R, Y};  
range Digits = 0..9;  
var{int} value[Letters] in Digits;  
var{int} carry[1..4] in 0..1;  
  
solve {  
    forall(i in Letters, j in Letters: i < j)  
        value[i] ≠ value[j];  
    value[S] ≠ 0;  
    value[M] ≠ 0;  
    carry[4] = value[M];  
    carry[3] + value[S] + value[M] = value[O] + 10 * carry[4];  
    carry[2] + value[E] + value[O] = value[N] + 10 * carry[3];  
    carry[1] + value[N] + value[R] = value[E] + 10 * carry[2];  
    value[D] + value[E] = value[Y] + 10 * carry[1];  
}
```

$$\begin{array}{r} C_4 \quad C_3 \quad C_2 \quad C_1 \\ S \quad E \quad N \quad D \\ + \quad M \quad O \quad R \quad E \\ \hline = \quad M \quad O \quad N \quad E \quad Y \end{array}$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
forall(i in Letters, j in Letters: i < j)
    value[i] ≠ value[j];
value[S] ≠ 0;
value[M] ≠ 0;
carry[4]
= value[M];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
forall(i in Letters, j in Letters: i < j)
    value[i] ≠ value[j];
value[S] ≠ 0;
value[M] ≠ 0;
carry[4]
= value[M];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
forall(i in Letters, j in Letters: i < j)
    value[i] ≠ value[j];
value[S] ≠ 0;
value[M] ≠ 0;
carry[4]
= value[M];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
forall(i in Letters, j in Letters: i < j)
    value[i] ≠ value[j];
value[S] ≠ 0;
value[M] ≠ 0;
carry[4]
= value[M];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
forall(i in Letters, j in Letters: i < j)
    value[i] ≠ value[j];
value[S] ≠ 0;
value[M] ≠ 0;
carry[4]
= value[M];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
forall(i in Letters, j in Letters: i < j)
    value[i] ≠ value[j];
value[S] ≠ 0;
value[M] ≠ 0;
carry[4]
= value[M];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
forall(i in Letters, j in Letters: i < j)
    value[i] ≠ value[j];
value[S] ≠ 0;
value[M] ≠ 0;
carry[4]
= value[M];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
forall(i in Letters, j in Letters: i < j)
    value[i] ≠ value[j];
value[S] ≠ 0;
value[M] ≠ 0;
carry[4]
= value[M];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red								
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
forall(i in Letters, j in Letters: i < j)
    value[i] ≠ value[j];
value[S] ≠ 0;
value[M] ≠ 0;
carry[4]
= value[M];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red								
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
forall(i in Letters, j in Letters: i < j)
    value[i] ≠ value[j];
value[S] ≠ 0;
value[M] ≠ 0;
carry[4] = value[M];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red								
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red								
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red								
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[3] + value[S] + value[M] = value[O] + 10 * carry[4];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	red	red								
E		red								
N		red								
D		red								
M	red	blue	red							
O		red								
R		red								
Y		red								
C <sub>4</sub>	red	blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[3] + value[S] + value[M] = value[O] + 10 * carry[4];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	■	■								
E		■								
N		■								
D		■								
M	■	■	■	■	■	■	■	■	■	■
O		■								
R		■								
Y		■								
C <sub>4</sub>	■	■								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[3] + value[S] + value[M] = value[O] + 10 * carry[4];
```

```
carry[3] + value[S] + 1 = value[O] + 10 * carry[4];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[3] + value[S] + value[M] = value[O] + 10 * carry[4];
```

```
carry[3] + value[S] + 1 = value[O] + 10 * carry[4];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[3] + value[S] + value[M] = value[O] + 10 * carry[4];
```

```
carry[3] + value[S] + 1 = value[O] + 10 * carry[4];
```

$\in [3..11]$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

carry[3] + value[S] + value[M] = value[O] + 10 \* carry[4];

carry[3] + value[S] + 1 = value[O] + 10 \* carry[4];

$$\begin{aligned} & \in [3..11] \\ & \in [0 + 2 + 1 .. 1 + 9 + 1] \end{aligned}$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[3] + value[S] + value[M] = value[O] + 10 * carry[4];
```

```
carry[3] + value[S] + 1 = value[O] + 10 * carry[4];
```

$$\in [0 + 2 + 1 .. 1 + 9 + 1]$$
$$\in [3..11]$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red								
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

carry[3] + value[S] + value[M] = value[O] + 10 \* carry[4];

carry[3] + value[S] + 1 = value[O] + 10 \* carry[4];

$$\in [0 + 2 + 1 .. 1 + 9 + 1]$$

$$\in [3..11]$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red								
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

carry[3] + value[S] + value[M] = value[O] + 10 \* carry[4];

carry[3] + value[S] + 1 = value[O] + 10 \* carry[4];

$$\in [0 + 2 + 1 .. 1 + 9 + 1]$$

$$\in [3..11]$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red								
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

carry[3] + value[S] + value[M] = value[O] + 10 \* carry[4];

carry[3] + value[S] + 1 = value[O] + 10 \* carry[4];

$$\in [0 + 2 + 1 .. 1 + 9 + 1]$$

$$\in [3..11]$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red								
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$\text{carry}[3] + \text{value}[S] + \text{value}[M] = \text{value}[O] + 10 * \text{carry}[4];$

$\text{carry}[3] + \text{value}[S] + 1 = \text{value}[O] + 10 * \text{carry}[4];$

$$\in [0 + 2 + 1 .. 1 + 9 + 1]$$

$$\in [3..11]$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red								
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$\text{carry}[3] + \text{value}[S] + \text{value}[M] = \text{value}[O] + 10 * \text{carry}[4];$

$\text{carry}[3] + \text{value}[S] + 1 = \text{value}[O] + 10 * \text{carry}[4];$

$$\in [0 + 2 + 1 .. 1 + 9 + 1]$$

$$\in [3..11]$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

carry[3] + value[S] + value[M] = value[O] + 10 \* carry[4];

carry[3] + value[S] + 1 = value[O] + 10 \* carry[4];

$$\begin{aligned} & \in [3..11] \\ & \in [0 + 2 + 1 .. 1 + 9 + 1] \end{aligned}$$

$$\in [10..19]$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red								
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

carry[3] + value[S] + value[M] = value[O] + 10 \* carry[4];

carry[3] + value[S] + 1 = value[O] + 10 \* carry[4];

$$\in [3..11]$$

$$\in [0 + 2 + 1 .. 1 + 9 + 1]$$

$$[3..11] \cap [10..19] = [10..11]$$

$$\in [10..19]$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

carry[3] + value[S] + value[M] = value[O] + 10 \* carry[4];

carry[3] + value[S] + 1 = value[O] + 10 \* carry[4];

10 <= carry[3] + value[S] + 1 <= 11

$\in [0 + 2 + 1 .. 1 + 9 + 1]$

[3..11]  $\cap$  [10..19] = [10..11]

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red								
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$10 \leq \text{carry}[3] + \text{value}[S] + 1 \leq 11$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red								
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$10 \leq \text{carry}[3] + \text{value}[S] + 1 \leq 11$

$10 - 1 \leq \text{carry}[3] + \text{value}[S] \leq 11 - 1$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$10 \leq \text{carry}[3] + \text{value}[S] + 1 \leq 11$

$10 - 1 \leq \text{carry}[3] + \text{value}[S] \leq 11 - 1$

$10 - 1 - \text{carry}[3] \leq \text{value}[S] \leq 11 - 1 - \text{carry}[3]$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red								
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$10 \leq \text{carry}[3] + \text{value}[S] + 1 \leq 11$

$10 - 1 \leq \text{carry}[3] + \text{value}[S] \leq 11 - 1$

$10 - 1 - \text{carry}[3] \leq \text{value}[S] \leq 11 - 1 - \text{carry}[3]$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red								
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$$10 \leq \text{carry}[3] + \text{value}[S] + 1 \leq 11$$

$$10 - 1 \leq \text{carry}[3] + \text{value}[S] \leq 11 - 1$$

$$10 - 1 - \text{carry}[3] \leq \text{value}[S] \leq 11 - 1 - \text{carry}[3]$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$10 \leq \text{carry}[3] + \text{value}[S] + 1 \leq 11$

$10 - 1 \leq \text{carry}[3] + \text{value}[S] \leq 11 - 1$

$10 - 1 - \text{carry}[3] \leq \text{value}[S] \leq 11 - 1 - \text{carry}[3]$

$10 - 1 - 1 \leq \text{value}[S] \leq 11 - 1 - 0$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$$10 \leq \text{carry}[3] + \text{value}[S] + 1 \leq 11$$

$$10 - 1 \leq \text{carry}[3] + \text{value}[S] \leq 11 - 1$$

$$10 - 1 - \text{carry}[3] \leq \text{value}[S] \leq 11 - 1 - \text{carry}[3]$$

$$10 - 1 - 1 \leq \text{value}[S] \leq 11 - 1 - 0$$

$$8 \leq \text{value}[S] \leq 10$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										
C <sub>4</sub>										
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$10 \leq \text{carry}[3] + \text{value}[S] + 1 \leq 11$

$10 - 1 \leq \text{carry}[3] + \text{value}[S] \leq 11 - 1$

$10 - 1 - \text{carry}[3] \leq \text{value}[S] \leq 11 - 1 - \text{carry}[3]$

$10 - 1 - 1 \leq \text{value}[S] \leq 11 - 1 - 0$

$8 \leq \text{value}[S] \leq 10$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red	Red		
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red	Red		
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O		Red								
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

carry[3] + value[S] + 1

= value[0] + 10 \* carry[4];

$\in [10..11]$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	■	■	■	■	■	■	■	■		
E		■								
N		■								
D		■								
M	■	■	■	■	■	■	■	■	■	■
O		■								
R		■								
Y		■								
C <sub>4</sub>	■	■								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

carry[3] + value[S] + 1 = value[0] + 10 \* carry[4];

10 <= value[0] + 10 \* carry[4] <= 11

[10...11]

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	■	■	■	■	■	■	■	■		
E		■								
N		■								
D		■								
M	■	■	■	■	■	■	■	■	■	
O		■								
R		■								
Y		■								
C <sub>4</sub>	■	■								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[3] + value[S] + 1 = value[0] + 10 * carry[4];
```

```
10 <= value[0] + 10 * carry[4] <= 11
```

[10...11]

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	■	■	■	■	■	■	■	■		
E		■								
N		■								
D		■								
M	■	■	■	■	■	■	■	■	■	■
O		■								
R		■								
Y		■								
C <sub>4</sub>	■	■								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[3] + value[S] + 1 = value[0] + 10 * carry[4];
```

```
10 <= value[0] + 10 * carry[4] <= 11
```

```
10 <= value[0] + 10 * 1 <= 11
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	■	■	■	■	■	■	■	■		
E		■								
N		■								
D		■								
M	■	■	■	■	■	■	■	■	■	■
O		■								
R		■								
Y		■								
C <sub>4</sub>	■	■								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

carry[3] + value[S] + 1 = value[0] + 10 \* carry[4];

10 <= value[0] + 10 \* carry[4] <= 11

10 <= value[0] + 10 \* 1 <= 11

0 <= value[0] <= 1

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	■	■	■	■	■	■	■	■		
E		■								
N		■								
D		■								
M	■	■	■	■	■	■	■	■	■	■
O		■	■	■	■	■	■	■	■	■
R		■								
Y		■								
C <sub>4</sub>	■	■								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

carry[3] + value[S] + 1 = value[0] + 10 \* carry[4];

10 <= value[0] + 10 \* carry[4] <= 11

10 <= value[0] + 10 \* 1 <= 11

0 <= value[0] <= 1

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red	Red		
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	Red
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

carry[3] + value[S] + 1 = value[0] + 10 \* carry[4];

10 <= value[0] + 10 \* carry[4] <= 11

10 <= value[0] + 10 \* 1 <= 11

0 <= value[0] <= 1

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red	Red		
E		Red								
N		Red								
D		Red								
M	Red	Blue	Red							
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	Red
R		Red								
Y		Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

carry[3] + value[S] + 1 = value[0] + 10 \* carry[4];

10 <= value[0] + 10 \* carry[4] <= 11

10 <= value[0] + 10 \* 1 <= 11

0 <= value[0] <= 1

forall(i in Letters, j in Letters)  
value[i] ≠ value[j];

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	■	■	■	■	■	■	■			
E	■	■								
N	■	■								
D	■	■								
M	■	■	■	■	■	■	■	■	■	■
O	■	■	■	■	■	■	■	■	■	■
R	■	■								
Y	■	■								
C <sub>4</sub>	■	■								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

carry[3] + value[S] + 1 = value[0] + 10 \* carry[4];

10 <= value[0] + 10 \* carry[4] <= 11

10 <= value[0] + 10 \* 1 <= 11

0 <= value[0] <= 1

forall(i in Letters, j in Letters)  
value[i] ≠ value[j];

# Send More Money

```
enum Letters = { S, E, N, D, M, O, R, Y};  
range Digits = 0..9;  
var{int} value[Letters] in Digits;  
var{int} carry[1..4] in 0..1;  
  
solve {  
    forall(i in Letters, j in Letters: i < j)  
        value[i] ≠ value[j];  
    value[S] ≠ 0;  
    value[M] ≠ 0;  
    carry[4] = value[M];  
    carry[3] + value[S] + value[M] = value[O] + 10 * carry[4];  
    [carry[2] + value[E] + value[O] = value[N] + 10 * carry[3];  
    `carry[1] + value[N] + value[R] = value[E] + 10 * carry[2];  
    value[D] + value[E] = value[Y] + 10 * carry[1];  
}
```

$$\begin{array}{r} C_4 \quad C_3 \quad C_2 \quad C_1 \\ S \quad E \quad N \quad D \\ + \quad M \quad O \quad R \quad E \\ \hline = \quad M \quad O \quad N \quad E \quad Y \end{array}$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	red	red	red	red	red	red	red	red		
E	red	red								
N	red	red								
D	red	red								
M	red	blue	red							
O	blue	red	red	red	red	red	red	red	red	red
R	red	red								
Y	red	red								
C <sub>4</sub>	red	blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[2] + value[E] + value[O] = value[N] + 10 * carry[3];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	red	red	red	red	red	red	red	red		
E	red	red								
N	red	red								
D	red	red								
M	red	blue	red							
O	blue	red	red	red	red	red	red	red	red	red
R	red	red								
Y	red	red								
C <sub>4</sub>	red	blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[2] + value[E] + value[O] = value[N] + 10 * carry[3];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	red	red	red	red	red	red	red			
E	red	red								
N	red	red								
D	red	red								
M	red	blue	red							
O	blue	red	red	red	red	red	red	red	red	red
R	red	red								
Y	red	red								
C <sub>4</sub>	red	blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[2] + value[E] + value[O] = value[N] + 10 * carry[3];  
carry[2] + value[E] + 0 = value[N] + 10 * carry[3];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red			
E	Red	Red								
N	Red	Red								
D	Red	Red								
M	Red	Blue	Red							
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	Red
R	Red	Red								
Y	Red	Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[2] + value[E] + value[O] = value[N] + 10 * carry[3];
```

```
carry[2] + value[E] + 0           = value[N] + 10 * carry[3];
```

$$\in [2..10]$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red			
E	Red	Red								
N	Red	Red								
D	Red	Red								
M	Red	Blue	Red							
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	Red
R	Red	Red								
Y	Red	Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

`carry[2] + value[E] + value[O] = value[N] + 10 * carry[3];`

`carry[2] + value[E] + 0 = value[N] + 10 * carry[3];`

$\in [2..10]$

$2 \leq value[N] + 10 * carry[3] \leq 10$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red	Red		
E	Red	Red								
N	Red	Red								
D	Red	Red								
M	Red	Blue	Red							
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	Red
R	Red	Red								
Y	Red	Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$2 \leq \text{value}[N] + 10 * \text{carry}[3] \leq 10$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red	Red		
E	Red	Red								
N	Red	Red								
D	Red	Red								
M	Red	Blue	Red							
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	Red
R	Red	Red								
Y	Red	Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$2 \leq \text{value}[N] + 10 * \text{carry}[3] \leq 10$

$2 - \text{value}[N] \leq 10 * \text{carry}[3] \leq 10 - \text{value}[N]$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red	Red		
E	Red	Red								
N	Red	Red								
D	Red	Red								
M	Red	Blue	Red							
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	Red
R	Red	Red								
Y	Red	Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$2 \leq \text{value}[N] + 10 * \text{carry}[3] \leq 10$

$2 - \text{value}[N] \leq 10 * \text{carry}[3] \leq 10 - \text{value}[N]$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red	Red		
E	Red	Red								
N	Red	Red								
D	Red	Red								
M	Red	Blue	Red							
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	Red
R	Red	Red								
Y	Red	Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$2 \leq \text{value}[N] + 10 * \text{carry}[3] \leq 10$

$2 - \text{value}[N] \leq 10 * \text{carry}[3] \leq 10 \rightarrow \text{value}[N]$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	red	red	red	red	red	red	red	red		
E	red	red								
N	red	red								
D	red	red								
M	red	blue	red							
O	blue	red	red	red	red	red	red	red	red	red
R	red	red								
Y	red	red								
C <sub>4</sub>	red	blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$$2 \leq \text{value}[N] + 10 * \text{carry}[3] \leq 10$$

$$2 - \text{value}[N] \leq 10 * \text{carry}[3] \leq 10 - \text{value}[N]$$

$$-7 \leq 10 * \text{carry}[3] \leq 8$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red			
E	Red	Red								
N	Red	Red								
D	Red	Red								
M	Red	Blue	Red							
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	Red
R	Red	Red								
Y	Red	Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$$2 \leq \text{value}[N] + 10 * \text{carry}[3] \leq 10$$

$$2 - \text{value}[N] \leq 10 * \text{carry}[3] \leq 10 - \text{value}[N]$$

$$-7 \leq 10 * \text{carry}[3] \leq 8$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	red	red	red	red	red	red	red	red		
E	red	red								
N	red	red								
D	red	red								
M	red	blue	red							
O	blue	red	red	red	red	red	red	red	red	red
R	red	red								
Y	red	red								
C <sub>4</sub>	red	blue								
C <sub>3</sub>										
C <sub>2</sub>										
C <sub>1</sub>										

$$2 \leq \text{value}[N] + 10 * \text{carry}[3] \leq 10$$

$$2 - \text{value}[N] \leq 10 * \text{carry}[3] \leq 10 - \text{value}[N]$$

$$-7 \leq 10 * \text{carry}[3] \leq 8$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	red	red	red	red	red	red	red	red		
E	red	red								
N	red	red								
D	red	red								
M	red	blue	red							
O	blue	red	red	red	red	red	red	red	red	red
R	red	red								
Y	red	red								
C <sub>4</sub>	red	blue								
C <sub>3</sub>			red							
C <sub>2</sub>										
C <sub>1</sub>										

$$2 \leq \text{value}[N] + 10 * \text{carry}[3] \leq 10$$

$$2 - \text{value}[N] \leq 10 * \text{carry}[3] \leq 10 - \text{value}[N]$$

$$-7 \leq 10 * \text{carry}[3] \leq 8$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red	Red		
E	Red	Red								
N	Red	Red								
D	Red	Red								
M	Red	Blue	Red							
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	Red
R	Red	Red								
Y	Red	Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>	Blue	Red								
C <sub>2</sub>										
C <sub>1</sub>										

$$2 \leq \text{value}[N] + 10 * \text{carry}[3] \leq 10$$

$$2 - \text{value}[N] \leq 10 * \text{carry}[3] \leq 10 - \text{value}[N]$$

$$-7 \leq 10 * \text{carry}[3] \leq 8$$

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red	Red		
E	Red	Red								
N	Red	Red								
D	Red	Red								
M	Red	Blue	Red							
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	Red
R	Red	Red								
Y	Red	Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>	Blue	Red								
C <sub>2</sub>										
C <sub>1</sub>										

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	red	red	red	red	red	red	red	red		
E	red	red								
N	red	red								
D	red	red								
M	red	blue	red							
O	blue	red	red	red	red	red	red	red	red	red
R	red	red								
Y	red	red								
C <sub>4</sub>	red	blue								
C <sub>3</sub>	blue	red								
C <sub>2</sub>										
C <sub>1</sub>										

```
enum Letters = { S, E, N, D, M, O, R, Y};  
range Digits = 0..9;  
var{int} value[Letters] in Digits;  
var{int} carry[1..4] in 0..1;  
  
solve {  
    forall(i in Letters, j in Letters: i < j)  
        value[i] ≠ value[j];  
    value[S] ≠ 0;  
    value[M] ≠ 0;  
    carry[4] = value[M];  
    carry[3] + value[S] + value[M] = value[O] + 10 * carry[4];  
    carry[2] + value[E] + value[O] = value[N] + 10 * carry[3];  
    carry[1] + value[N] + value[R] = value[E] + 10 * carry[2];  
    value[D] + value[E] = value[Y] + 10 * carry[1];  
}
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red	Red		
E	Red	Red								
N	Red	Red								
D	Red	Red								
M	Red	Blue	Red							
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	Red
R	Red	Red								
Y	Red	Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>	Blue	Red								
C <sub>2</sub>										
C <sub>1</sub>										

```

enum Letters = { S, E, N, D, M, O, R, Y};
range Digits = 0..9;
var{int} value[Letters] in Digits;
var{int} carry[1..4] in 0..1;

solve {
    forall(i in Letters, j in Letters: i < j)
        value[i] ≠ value[j];
    value[S] ≠ 0;
    value[M] ≠ 0;
    carry[4] = value[M];
    carry[3] + value[S] + value[M] = value[O] + 10 * carry[4];
    carry[2] + value[E] + value[O] = value[N] + 10 * [carry[3]];
    carry[1] + value[N] + value[R] = value[E] + 10 * carry[2];
    value[D] + value[E] = value[Y] + 10 * carry[1];
}

```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	red	red	red	red	red	red	red	red		
E	red	red								
N	red	red								
D	red	red								
M	red	blue	red							
O	blue	red	red	red	red	red	red	red	red	red
R	red	red								
Y	red	red								
C <sub>4</sub>	red	blue								
C <sub>3</sub>	blue	red								
C <sub>2</sub>										
C <sub>1</sub>										

```
enum Letters = { S, E, N, D, M, O, R, Y};  
range Digits = 0..9;  
var{int} value[Letters] in Digits;  
var{int} carry[1..4] in 0..1;  
  
solve {  
    forall(i in Letters, j in Letters: i < j)  
        value[i] ≠ value[j];  
    value[S] ≠ 0;  
    value[M] ≠ 0;  
    carry[4] = value[M];  
    [carry[3]] + value[S] + value[M] = value[O] + 10 * carry[4];  
    carry[2] + value[E] + value[O] = value[N] + 10 * [carry[3]];  
    carry[1] + value[N] + value[R] = value[E] + 10 * carry[2];  
    value[D] + value[E] = value[Y] + 10 * carry[1];  
}
```

# Computational Paradigm

- ▶ The propagation engine
  - this is the core of any constraint-programming solver
  - a simple (fixpoint) algorithm

```
propagate()
{
    repeat
        select a constraint c;
        if c is infeasible given the domain store then
            return failure;
        else
            apply the pruning algorithm associated with c;
    until no constraint can remove any value from the
    domain of its variables;
    return success;
}
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	red	red	red	red	red	red	red	red		
E	red	red								
N	red	red								
D	red	red								
M	red	blue	red							
O	blue	red	red	red	red	red	red	red	red	red
R	red	red								
Y	red	red								
C <sub>4</sub>	red	blue								
C <sub>3</sub>	blue	red								
C <sub>2</sub>										
C <sub>1</sub>										

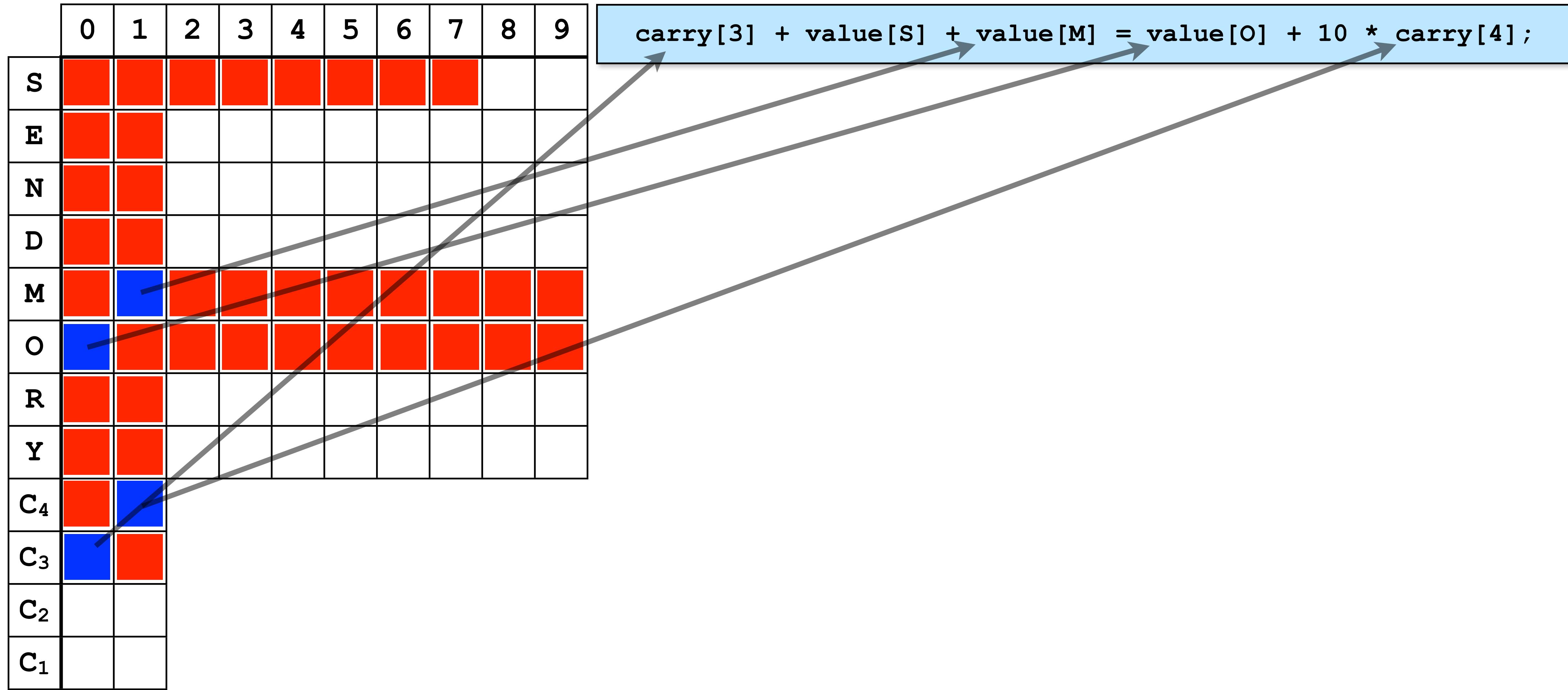
```
enum Letters = { S, E, N, D, M, O, R, Y};  
range Digits = 0..9;  
var{int} value[Letters] in Digits;  
var{int} carry[1..4] in 0..1;  
  
solve {  
    forall(i in Letters, j in Letters: i < j)  
        value[i] ≠ value[j];  
    value[S] ≠ 0;  
    value[M] ≠ 0;  
    carry[4] = value[M];  
    [carry[3]] + value[S] + value[M] = value[O] + 10 * carry[4];  
    carry[2] + value[E] + value[O] = value[N] + 10 * [carry[3]];  
    carry[1] + value[N] + value[R] = value[E] + 10 * carry[2];  
    value[D] + value[E] = value[Y] + 10 * carry[1];  
}
```

# What is the Search Space?

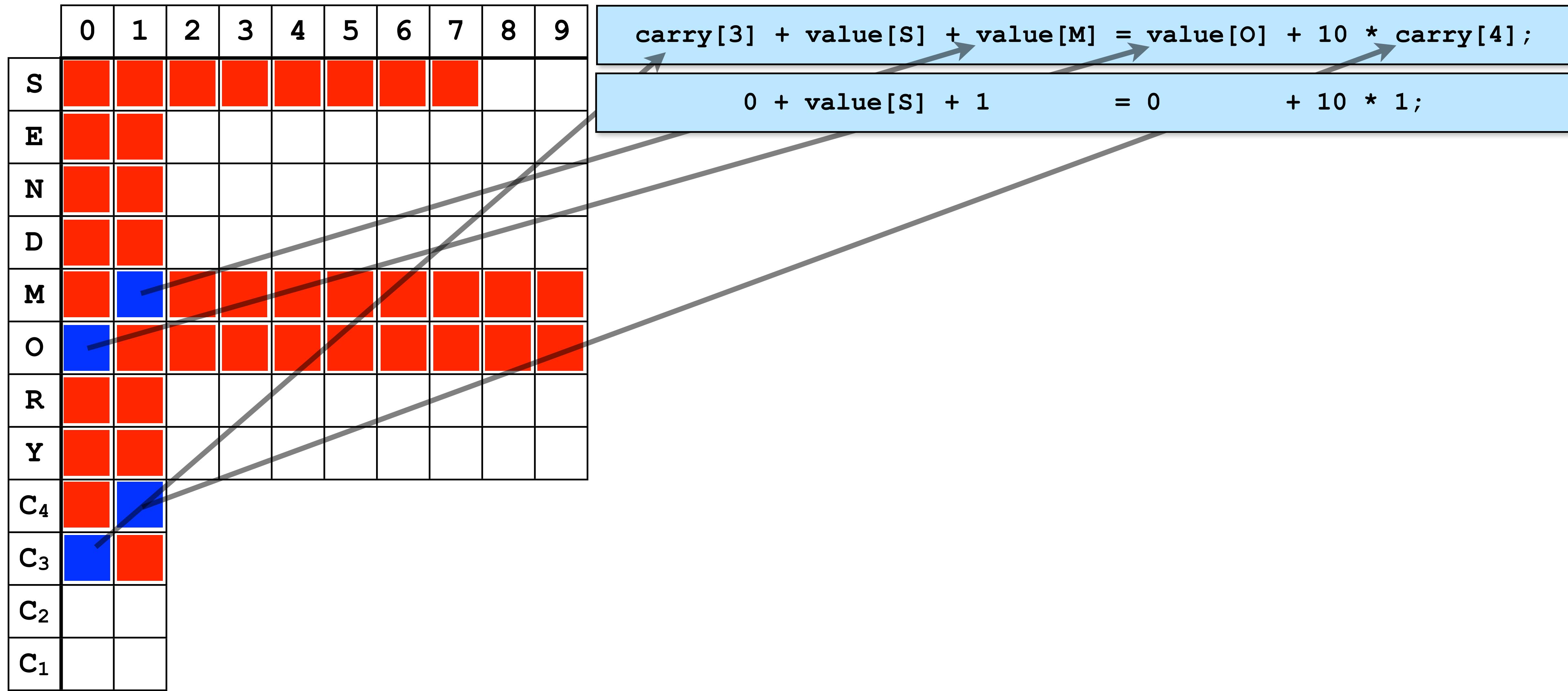
	0	1	2	3	4	5	6	7	8	9
S	red	red	red	red	red	red	red	red		
E	red	red								
N	red	red								
D	red	red								
M	red	blue	red							
O	blue	red	red	red	red	red	red	red	red	red
R	red	red								
Y	red	red								
C <sub>4</sub>	red	blue								
C <sub>3</sub>	blue	red								
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[3] + value[S] + value[M] = value[O] + 10 * carry[4];
```

# What is the Search Space?



# What is the Search Space?



# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red			
E	Red	Red								
N	Red	Red								
D	Red	Red								
M	Red	Blue	Red							
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	Red
R	Red	Red								
Y	Red	Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>	Blue	Red								
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[3] + value[S] + value[M] = value[O] + 10 * carry[4];
```

```
0 + value[S] + 1 = 0 + 10 * 1;
```

```
value[S] = 9;
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red	Red	Blue	
E	Red	Red								
N	Red	Red								
D	Red	Red								
M	Red	Blue	Red	Red						
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	Red
R	Red	Red								
Y	Red	Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>	Blue	Red								
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[3] + value[S] + value[M] = value[O] + 10 * carry[4];
```

```
0 + value[S] + 1 = 0 + 10 * 1;
```

```
value[S] = 9;
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red	Red	Blue	
E	Red	Red								
N	Red	Red								
D	Red	Red								
M	Red	Blue	Red	Red						
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	Red
R	Red	Red								
Y	Red	Red								
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>	Blue	Red								
C <sub>2</sub>										
C <sub>1</sub>										

```
carry[3] + value[S] + value[M] = value[O] + 10 * carry[4];
```

```
0 + value[S] + 1 = 0 + 10 * 1;
```

```
value[S] = 9;
```

```
forall(i in Letters, j in Letters: i < j)
    value[i] ≠ value[j];
```

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9	
S	Red	Red	Red	Red	Red	Red	Red	Red	Blue		carry[3] + value[S] + value[M] = value[O] + 10 * carry[4];
E	Red	Red							Red		0 + value[S] + 1 = 0 + 10 * 1;
N	Red	Red							Red		value[S] = 9;
D	Red	Red							Red		forall(i in Letters, j in Letters: i < j) value[i] ≠ value[j];
M	Red	Blue	Red								
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red		
R	Red	Red							Red		
Y	Red	Red							Red		
C <sub>4</sub>	Red	Blue									
C <sub>3</sub>	Blue	Red									
C <sub>2</sub>											
C <sub>1</sub>											

# What is the Search Space?

	0	1	2	3	4	5	6	7	8	9
S	Red	Red	Red	Red	Red	Red	Red	Red	Blue	
E	Red	Red							Red	
N	Red	Red							Red	
D	Red	Red						Red		
M	Red	Blue	Red							
O	Blue	Red	Red	Red	Red	Red	Red	Red	Red	
R	Red	Red							Red	
Y	Red	Red							Red	
C <sub>4</sub>	Red	Blue								
C <sub>3</sub>	Blue	Red								
C <sub>2</sub>										
C <sub>1</sub>										

# Linear Constraints over Integers

- Consider a constraint

$$a_1x_1 + \dots + a_nx_n \geq b_1y_1 + \dots + b_my_m$$

$a_i, b_j \geq 0$  are constants

$x_i, y_j$  are variables with domains  $D(x_i), D(y_j)$

# Linear Constraints over Integers

- Consider a constraint

$$a_1x_1 + \dots + a_nx_n \geq b_1y_1 + \dots + b_my_m$$

$a_i, b_j \geq 0$  are constants

$x_i, y_j$  are variables with domains  $D(x_i), D(y_j)$

- Feasibility test

# Linear Constraints over Integers

- Consider a constraint

$$a_1x_1 + \dots + a_nx_n \geq b_1y_1 + \dots + b_my_m$$

$a_i, b_j \geq 0$  are constants

$x_i, y_j$  are variables with domains  $D(x_i), D(y_j)$

- Feasibility test

$$a_1 \max(D(x_1)) + \dots + a_n \max(D(x_n)) \geq b_1 \min(D(y_1)) + \dots + b_m \min(D(y_m))$$

# Linear Constraints over Integers

- Consider a constraint

$$a_1x_1 + \dots + a_nx_n \geq b_1y_1 + \dots + b_my_m$$

$a_i, b_j \geq 0$  are constants

$x_i, y_j$  are variables with domains  $D(x_i), D(y_j)$

- Feasibility test

$$a_1 \max(D(x_1)) + \dots + a_n \max(D(x_n)) \geq b_1 \min(D(y_1)) + \dots + b_m \min(D(y_m))$$

$$l = a_1 \max(D(x_1)) + \dots + a_n \max(D(x_n))$$

$$r = b_1 \min(D(y_1)) + \dots + b_m \min(D(y_m))$$

# Linear Constraints over Integers

- Consider a constraint

$$a_1x_1 + \dots + a_nx_n \geq b_1y_1 + \dots + b_my_m$$

$a_i, b_j \geq 0$  are constants

$x_i, y_j$  are variables with domains  $D(x_i), D(y_j)$

- Pruning

$$a_i x_i \geq r - (l - a_i \max(D(x_i)))$$

$$x_i \geq \left\lceil \frac{r - (l - a_i \max(D(x_i)))}{a_i} \right\rceil$$

$$y_j \leq \left\lfloor \frac{l - (r - b_j \min(D(y_j)))}{b_j} \right\rfloor$$

# Until Next Time