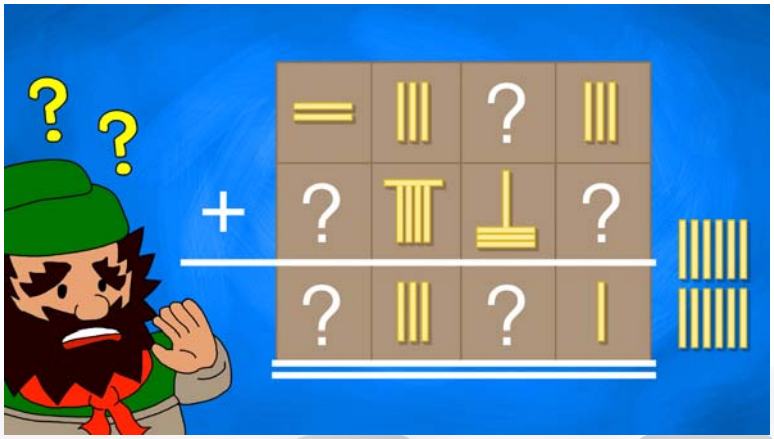
全局约束

李浩文、彼得·斯塔基

张飞算账

张飞算账

- 12根算筹被打乱
- 被打乱的数字是不同的



3

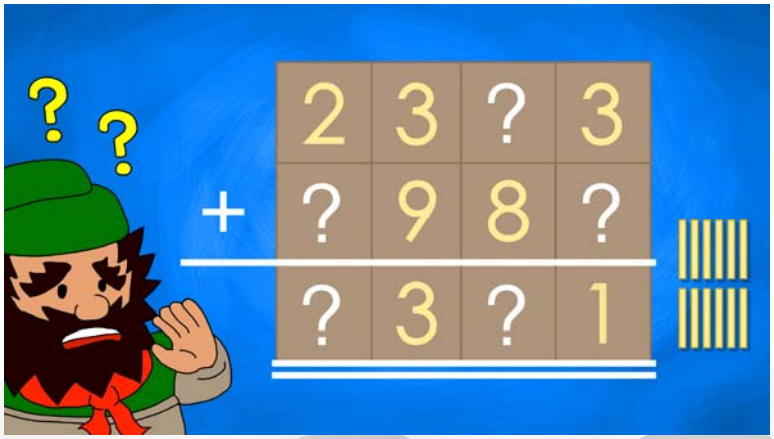
算筹对应表示

	0	1	2	3	4	5	6	7	8	9
直式										
横式										

4

张飞算账

- 12根算筹被打乱
- 被打乱的数字是不同的



5

算筹问题 (rods.mzn)

```
set of int: DIGIT = 1..9;
array[DIGIT] of int: rods = [1,2,3,4,5,2,3,4,5];

var DIGIT: M1; % first messed up digit
var DIGIT: M2; % second messed up digit
var DIGIT: M3; % third messed up digit
var DIGIT: M4; % fourth messed up digit
var DIGIT: M5; % fifth messed up digit
```

（根据）变量查找

```
constraint rods[M1] + rods[M2] + rods[M3] +
           rods[M4] + rods[M5] = 12;

constraint 2303 + M1 * 10 + 980 + M2 * 1000 + M3
           = 301 + M4 * 1000 + M5 * 10;
```

6

算筹问题（续）(rods.mzn)

```
% alldifferent messed up digits
constraint M1 != M2;
constraint M1 != M3;
constraint M1 != M4;
constraint M1 != M5;
constraint M2 != M3;
constraint M2 != M4;
constraint M2 != M5;
constraint M3 != M4;
constraint M3 != M5;
constraint M4 != M5;

% alldifferent messed up digits
include "alldifferent.mzn";

constraint alldifferent([M1,M2,M3,M4,M5]);

solve satisfy;
```

冗长乏味

include

全局约束

7

集合参数和变量对应表

- 集合参数可如下定义
  - set of 类型: 变量名 = 固定值集合;
  - 它们可以作为固定值集合使用
- 变量对应表
  - 下标表达式可以包括决策变量
  - 这给了我们很强的建模（描述）能力
  - 这里我们只是简单地使用
  - 以后会有更多例子

8

Include

- Include 语句
  - `include "文件名";`
  - 会在MiniZinc模型中包含该文件
  - 会在当前文件夹和MiniZinc库路径中寻找
- Include 语句用于
  - 将较大模型分成各个小部分
  - 加载全局约束的定义
  - 控制MiniZinc面对某一求解器时应加载哪个全局约束的定义

9

全局约束

- 原则上，任何可以取无限个变量为参数的约束（就是全局约束）
  - 所以线性约束是“全局的”
- 全局约束是
  - 在很多问题都会出现的约束
- 全局约束可以使
  - 模型更简短
  - 求解变得更容易（因为求解器可以利用全局约束的结构信息）

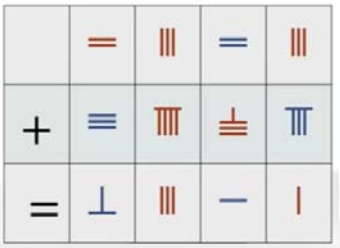
10

算筹问题

- **求解模型得到**

```
M1 = 2;
M2 = 3;
M3 = 8;
M4 = 6;
M5 = 1;
```

- 问题的解就是



11

小结

- **全局**约束是我们在课程会遇到的最重要有效**的概念之一**
  - **之后**还有很多相关知识
- 类似地，允许在数组的下标表达式中使用决**策**变量也是一个非常有效的建模工具

12

图像引用

所有图像由Marti Wong设计提供, © 香港中文大学与墨尔本大学 2016

13