



第三个模型

李浩文、彼得·斯塔基



组建一支更强的军队



给地图着色



3

地图着色问题 (color.mzn)

```
enum COLOR = {GREEN, BLUE, PINK, YELLOW};
```

```
var COLOR: Si;  
var COLOR: Yan;  
var COLOR: Yu;  
var COLOR: Xu;  
var COLOR: Qing;  
var COLOR: Ji;  
var COLOR: You;  
var COLOR: Bing;  
var COLOR: Yong;  
var COLOR: Liang;  
var COLOR: Yi;  
var COLOR: Jing;  
var COLOR: Yang;  
var COLOR: Jiao;
```



4

地图着色问题 (color.mzn)

```
constraint Liang != Yong;  
constraint Yong != Yi;  
constraint Yong != Jing;  
constraint Yong != Si;  
constraint Yi != Jing;  
constraint Yi != Jiao;  
constraint Jiao != Jing;  
constraint Jiao != Yang;  
constraint Jing != Yang;  
constraint Jing != Si;  
constraint Jing != Yu;  
constraint Yang != Yu;  
constraint Yang != Xu;  
constraint Yu != Si;  
constraint Yu != Yan;  
constraint Yu != Xu;  
constraint Xu != Yan;
```

```
constraint Xu != Qing;  
constraint Yan != Si;  
constraint Yan != Ji;  
constraint Yan != Ji;  
constraint Yan != Qing;  
constraint Qing != Ji;  
constraint Ji != You;  
constraint Ji != Bing;  
constraint Ji != Si;  
constraint You != Bing;  
constraint Bing != Si;  
solve satisfy;
```



5

地图着色问题

- 我们可以运行如下命令求解MiniZinc模型

```
$ minizinc color.mzn
```

- 运行结果为

```
Si = PINK;
```

```
Yan = YELLOW;
```

```
Yu = BLUE;
```

```
Xu = GREEN;
```

```
...
```

- 每个州都被涂上一个和其相邻的州不同的颜色



6

枚举类型

- ⌘ 枚举类型 (enums) 定义一个具有有限对象的集合
 - 决策变量和参数可以是枚举类型
 - 数组下标 (之后介绍) 可以是枚举类型
 - 集合 (之后介绍) 可以基于枚举类型来定义
- ⌘ 枚举类型可以如下声明
 - `enum` 枚举类型名
- ⌘ 枚举类型可以由一个标识符集合来定义
 - 枚举类型名 = { `id1`, ..., `idn` } ;
- ⌘ 我们可以使用枚举类型名来声明变量
 - `[var]` 枚举类型名: 变量名

7

地图着色问题

- ⌘ 如果我们更改程序 (使用枚举类型)
 - `COLOR = {GREEN, BLUE, PINK};`
- ⌘ 求解MiniZinc模型

```
$ minizinc color.mzn
```
- ⌘ 运行结果为

```
====UNSATISFIABLE====
```
- ⌘ 这表明此问题没有解

8



小结

枚举类型

- 引入了一个已命名对象的集合
- 有助于模型的“类型安全性”
- 之后会有更多关于枚举类型的内容

不可满足的模型

- 并不是每个模型都有一个解！

图（地图）着色问题

- 一个经典的图论问题
- 在寄存器分配，时间表问题都有应用
- 纯图着色问题最好用专门的算法来解决

9

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

10