



# 对象建模

李浩文、彼得·斯塔基



## 桃园宴会



## 桃园宴会

- 为了出兵前振奋士气，刘备，关羽和张飞安排了一个宴会
- 他们想要用他们最喜爱的三种菜摆满桌子，来最大化宴会满足程度

			
	29	19	8
	8	5	3

3

## 桃园宴会

- 由于桌子的容量是18，根据已有数据所得的模型为

			
	29	19	8
	8	5	3








$$\begin{aligned} &\text{maximize} && 29S + 19F + 8M \\ &\text{subject to} && 8S + 5F + 3M \leq 18 \end{aligned}$$

4

## 桃园宴会

- 他们也为士兵们准备了一个宴会，提供了不同的菜品和一个更大的桌子

- 容量70

					
	18	16	14	13	6
	12	10	9	8	4

- 我们想要建一个对不同规模的数据可以重复使用的模型！

5

## 桃园宴会 (banquet.mzn)

```
enum DISH;  
int: capacity;  
array[DISH] of int: satisf;  
array[DISH] of int: size;
```

数组声明

```
array[DISH] of var int: amt; % how many of each dish
```

数组查找

```
constraint forall(i in DISH)(amt[i] >= 0);  
constraint sum(i in DISH)(size[i] * amt[i])  
    <= capacity;  
solve maximize sum(i in DISH)(satisf[i] * amt[i]);  
output ["Amount = ", show(amt), "\n"];
```

forall表达式

sum表达式

6

## MiniZinc的新特性

- ⌘ 下标范围表达式
  - `l..u` (`l`, `u`为整数) 或
  - 枚举类型
- ⌘ 参数和变量数组
  - `array[范围] of 变量声明`
- ⌘ 数组查找
  - 数组名[下标表达式]
- ⌘ 生成器表达式
  - `forall(i in 范围)(布尔型表达式)`
    - 对于范围内所有的*i*, 对应布尔型表达式都为真
  - `sum(i in 范围)(表达式)`
    - 对范围内所有*i*对应的表达式累加

7

## 数据文件 (banquet1.dzn)

```
DISH = {SNAKESOUP, GONGBAOFROGS, MAPOTOFU};  
capacity = 18;  
satisf = [29,19,8];  
size = [8,5,3];
```

```
$ minizinc banquet.mzn banquet1.dzn
```

```
amt = [1, 2, 0]  解  
-----      找到解  
=====      证明为最优解
```

8

## 数据文件 (banquet2.dzn)

```
DISH = {CHILIFISHHEAD, SAUSAGE, SEACUCUMBER,  
        CHICKEN, FRIEDRICE};  
capacity = 70;  
satisf = [18,16,14,13,6];  
size = [12,10,9,8,4];
```

```
$ minizinc banquet.mzn banquet2.dzn
```

```
amt = [0, 1, 0, 7, 1]  
----- 找到解  
===== 证明为最优解
```

当然，你也可以选择使用IDE！

9

## 对象建模

- 创建一个枚举类型来为对象命名，例如，  
DISH
- 创建一个参数数组来表示对象的每一个属性，  
例如，size, satisf
- 创建一个变量数组来表达对象的每一个决策，  
例如，amt
- 使用生成器表达式/推导式来建立限制对象的约束
- 注意，一个模型可能含有多个对象集合

10



## 小结

- ⌘ 枚举类型用来表示对象集合
- ⌘ 定义在对象集合上的数组用来表示
  - 对象特征
  - 对象决策
- ⌘ 生成器表达式
  - forall和sum
  - 用于构造表达式（约束多个对象）
- ⌘ 桃园宴会问题是众所周知的背包问题的一个版本

11

## 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

12