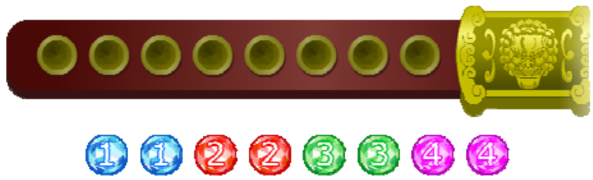
多重建模：排列

李浩文、彼得·斯塔基

衣带密诏问题

除非另有声明，本材料为©墨尔本大学与香港中文大学 版权所有。只能在个人参考、研究或学习的情况下，方可分享、复印或下载本材料。

衣带密诏问题



3

间隔约束

4

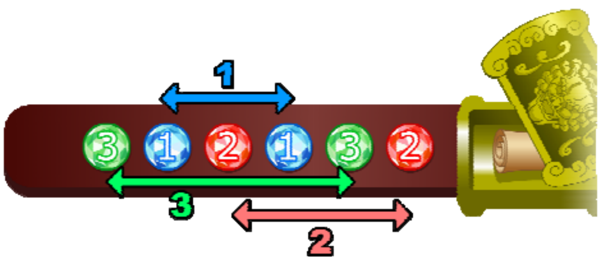
除非另有声明，本材料为©墨尔本大学与香港中文大学 版权所有。只能在个人参考、研究或学习的情况下，方可分享、复印或下载本材料。

间隔约束

5

衣带密诏问题举例

- B(*m*,*n*): 给定 1..*n* 中的每个数字 *m* 份， **找到**
  这些数字的一个序列，其中数字 *k* **的任何两**
  个连续出现的位置之间有 *k* **个数字**



6

除非另有声明，本材料为©墨尔本大学与香港中文大学 版权所有。只能在个人参考、研究或学习的情况下，方可分享、复印或下载本材料。

一个分配问题

- 这是一个怎样的分配问题？
- 映射 DOM = $1_1, 1_2, 2_1, 2_2, 3_1, 3_2, 4_1, 4_2$
  - 数字的有序副本
- 到 COD = 1,2,3,4,5,6,7,8
  - 序列中的位置

7

衣带密诏问题模型 (beltPos.mzn)

- 数据和决策变量

```
int: n;
set of int: DIG = 1..n;
int: m;
set of int: COPY = 1..m;
int: l = m*n;
set of int: POS = 1..l;
array[DIG,COPY] of var POS: po;
```

- 约束

```
forall(d in DIG, c in 1..m-1)
   (po[d,c+1] = po[d,c] + d + 1);
alldifferent([po[d,c] |
   d in DIG, c in COPY]);
```

8

除非另有声明，本材料为©墨尔本大学与香港中文大学 版权所有。只能在个人参考、研究或学习的情况下，方可分享、复印或下载本材料。

衣带密诏问题的逆向视角模型

- DOM代表位置，COD代表数字副本
- 我们需要映射DIG x COPY到一个整数: $d_c = m*(d-1) + c$
  - $1_1, 1_2, 2_1, 2_2, 3_1, 3_2, 4_1, 4_2 = 1, 2, 3, 4, 5, 6, 7, 8$

```
set of int: DIGCOP = 1..l;
array[POS] of var DIGCOP: dc;
```

- 从逆向视角出发，alldifferent约束很容易表达。

```
alldifferent([dc[p] | p in POS]);
```

9

逆向间隔约束

- 我们如何在逆向视角中表达间隔约束？
  - $dc[p] = d_c \iff po[d,c] = p$

```
forall(d in DIG, c in 1..m-1)
   (po[d,c+1] = po[d,c] + d + 1);
```

  - 因此，在逆向视觉中对应的约束表达为

```
forall(d in DIG, c in 1..m-1,
       p in POS)
   (dc[p] = m*(d-1) + c <->
       dc[p+d+1] = m*(d-1) + c + 1);
```

- 糟糕的表述
  - 如果位置p有d_c，则p+d+1有d_{c+1}

10

除非另有声明，本材料为©墨尔本大学与香港中文大学 版权所有。只能在个人参考、研究或学习的情况下，方可分享、复印或下载本材料。

关于逆向视角中的约束表达

```
forall(d in DIG, c in 1..m-1,
        p in POS)
     (dc[p] = m*(d-1) + c <->
        dc[p+d+1] = m*(d-1) + c + 1);
```

- 注意， 我们在访问超出 *dc* 数组的位置，例如，$l + d + 1$
- 但这不会引起问题
  - 除了数字d_m 的最后一个副本可以出现在序列的最后d个位置之外， 其他的都不可以
  - 关系语义要求当 $i > l$ 时，$dc[i] = j$ 的值为false, 因为 ***dc[i]* 是不存在的**

11

关于逆向视角中的约束表达

- 避免超出数组访问是更安全的
  - 但是在这个实例中显得笨拙

```
forall(d in DIG, c in 1..m-1,
      p in POS)
   (dc[p] = m*(d-1) + c <->
       if p+d+1 in POS then
          dc[p+d+1] = m*(d-1) + c + 1
       else false endif);
```

12

除非另有声明，本材料为©墨尔本大学与香港中文大学 版权所有。只能在个人参考、研究或学习的情况下，方可分享、复印或下载本材料。

衣带密诏问题的逆向视角模型 (beltDig.mzn)

```
include "globals.mzn";
int: n;
set of int: DIG = 1..n;
int: m;
set of int: COPY = 1..m;
int: l = m*n;
set of int: POS = 1..l;
set of int: DIGCOP = 1..l;

array[POS] of var DIGCOP: dc;

constraint forall(d in DIG, c in 1..m-1,
      p in POS)
   (dc[p] = m*(d-1) + c <->
      dc[p+d+1] = m*(d-1) + c + 1);
constraint alldifferent([dc[p] | p in POS]);

solve satisfy;
```

13

两个不同的模型

- 衣带密诏问题：
  - $po[d,c] = d_c$的位置
- 运用逆向视角的衣带密诏问题
  - $dc[p]$ = 数字d_c在位置p
- 哪个运行的更快？
- 哪个更容易输出宝石序列？

```
output["\((dc[p]-1) div m + 1) " |
   p in POS];
4 1 3 1 2 4 3 2
```

14

除非另有声明，本材料为©墨尔本大学与香港中文大学 版权所有。只能在个人参考、研究或学习的情况下，方可分享、复印或下载本材料。

结合的衣带密诏模型

- 我们可以结合模型
  - 省略alldifferent约束
    - 它们被inverse隐含了
  - 省略逆向视角中约束的表达
    - 它们被关于*po*[*d*,*c*]的等式隐含了
- CP求解器可以更好地求解结合的模型
  - 同时搜索*po*和*dc*变量

15

结合的衣带密诏模型 (beltComb.mzn)

```
include "globals.mzn";
int: n;
set of int: DIG = 1..n;
int: m;
set of int: COPY = 1..m;
int: l = m*n;

set of int: POS = 1..l;
array[DIG,COPY] of var POS: po;
set of int: DIGCOP = 1..l;
array[POS] of var DIGCOP: dc;

constraint forall(d in DIG, c in 1..m-1)
   (po[d,c+1] = po[d,c] + d + 1);
constraint inverse(dc,
   [po[d,c]|d in DIG, c in COPY]);

solve satisfy;
output["\((dc[p]-1) div m + 1) " | p in POS];
```

16

除非另有声明，本材料为©墨尔本大学与香港中文大学 版权所有。只能在个人参考、研究或学习的情况下，方可分享、复印或下载本材料。

小结

- 进一步阐释从不同视角来建模的例子
- **衣**带密诏问题是众所周知的 Langford **数列**问题的一个改编和泛化。Langford **数列**问题是**一个**应用在电路设计以及其他问题中的数学谜题
- **尽管从任何一个角度都可以完全描述例子中的需求，从特定的角度描述一些需求会更自然**

17

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

18

除非另有声明，本材料为©墨尔本大学与香港中文大学 版权所有。只能在个人参考、研究或学习的情况下，方可分享、复印或下载本材料。