**产生式系统三部分组成：**

* **综合数据库**
* **产生式规则集**
* **控制系统**

**综合数据库：存放问题的状态描述的数据结构。**

**控制系统：**

**（1）选择规则：**

**对同一个状态的多个可用规则排序。**

**（2）检验状态描述是否满足终止条件。**

**如果满足终止条件，则终止产生式系统的运行，**

**并用使用过的规则序列构造出问题的解。**

**产生式系统的基本过程**

**Procedure PRODUCTION**

1. **DATA←初始状态描述**
2. **until DATA 满足终止条件，do：**
3. **begin**
4. **在规则集合中，选出一条可用于DATA的规则R**
5. **DATA←把R应用于DATA所得的结果**
6. **End**

**选取规则所依据的原则称为控制策略。**

**高效的系统需要被求解问题足够的知识，以便在步骤4选出一条最合用的规则。**

**产生式系统的控制策略分为两类：**

* **不可撤回的控制策略**
* **试探性控制策略：回溯方式和图搜索方式**

**八数码难题 不可撤回式控制 P17 的疑惑**

**Q:为什么不按照左上右下的顺序而采用实例中的顺序？**

**不可撤回控制策略的优点**

**1. 只记录当前一个节点，空间复杂性很低。**

**2. 若能找到解，则速度很快**

**不可撤回控制策略的局限性**

**多数情况下找不到解。**

**原因：**

**(a) 爬山函数有多个局部极大值**

**(b) 爬山函数具有“平顶值”**

**Q:什么是“平顶值”？**

**解决方法：设计更好的爬山函数；选多余规则**

**Q:什么是多余规则？**

**回溯方式是一种试探性的控制策略。（类似深度优先）**

**图搜索控制策略(类似广度优先搜索策略)**

**产生式系统的工作方式**

* **正向产生式系统（数据驱动控制）**
* **反（逆）向产生式系统（目标驱动控制）**
* **双向产生式系统：正向产生式系统和反向产生式系统结合.**

**可交换产生式系统**

**可交换产生式系统,可应用不可撤回式控制策略**

**Q：为什么？**

**A：个人认为：已八数码难题为例，只要你找到能使爬山函数CF不减少，比如出现数个下一个状态的CF值相同，则可以任取一个。**

**可交换产生式系统性质（看C2-Slider 34,帮助理解性质，个人感觉其实就是多了一个节点的自循环）**

**可应用性：（自己理解）任意一条规则r ∈RD** ，可应用于D得到D’（ **D’= r** **(D)**）,则r可应用于D’（ **r∈ RD’ ，即RD⊆ RD’**）

**可满足性：（不理解）C2-Slider 32**

**Q：这里的目标条件是什么？**

**无次序性：（去C2-Slider 33看一下概念）**

**Note：产生式系统的可交换性并不意味着整个规则序列的次序可以改变。只是最初作用于给定状态的那些规则使用起来与次序无关。**

**可交换产生式系统可以避免搜索多余路径，可以使用不可撤回策略。**

**避免搜索多余路径的另一种方法是可分解的产生式系统。**

**重写问题的与/或树（看一下，练一下 C2-Slider 43）**