在SR中，执行是由一个**逻辑时钟控制**的，在时钟的每一个滴答中，角色在概念上**同时和瞬间**执行。SR是一种协调语言，而不是一种编程语言。这允许系统中的“基元”是复杂的组件，而不是内置的语言基元。这反过来又使MoCs(Model of Computations,计算模型s)主操作控制系统的不同种组合成为可能，因为复杂组件本身可能包括在另一种计算模型下开发的组件。

**关键点：**

**①角色、导演、信号**

**②层次化结构建模**

**③不动点：**

在一个构造良好的SR模型中，在有限的步骤中会有一个唯一的不动点。

s(n) = (f(n))(s(n))

模型中一直不变的值？1997第三章

滴答为n时，模型中的若干角色不断fire，只为找到不动点，然后调用postfire。

**④prefire，fire，postfire：**

角色的一次迭代被分成这三种动作。

prefire 测试角色调用所需的先决条件是否充足。

fire 角色的主要动作期，即读取输入、执行计算并生成输出数据。

postfire 更新角色的状态以响应后续输入，这是角色抽象语义的一个重要部分。

**⑤因果关系问题**

**⑥严格角色、非严格角色**

**⑦⊥：**

未知，unknown。flat CPOs.

在每一个滴答时，给所有信号赋值⊥，导演就可以按照任意顺序调用角色。

对于严格角色，如果有任何未知输入，则输出也必须为未知。

对于非严格角色，如果有一些输入是未知的，输出可能是已知的。  
  
**⑧单调：**

如果一个角色在给出更多输入信息后不改变它的输出信息，那它就是单调的

大多数Ptolemy II角色遵从严格角色语义并且是单调的，因此他们可以被用在SR。

逻辑时间（logical time）≠ 离散时间（discrete time ） 逻辑时间不像离散时间那样阶段之间有明显的时间间隔。

遗漏的 → FSM（挑重点看）→ SR语义（1997 3.3节）→ 1997第四章（了解所说的SR模型运行的两个策略是如何运行）

遗漏：

2014 5.2的例子仔细看

2014书中定义

2014 5.3前两段例子：①顺序执行，每个角色被fire一次②赋值⊥，随便执行

FSM：

2014 6.1,6.2.1~6.2.4,6.4

2008年 第三章

SR模型到Kripke 2008-C.P.Cheng-Applied Verification\_The\_Ptolemy\_Approach 第三章