第五章

- **谓词/变迁系统的定义**

定义 4.3 谓词/变迁网系统

九元组$\Sigma=(P,T;F,D,V,A_P,A_T,A_F,M_0)$是$P_r/T$系统的条件是：

1. $(P,T;F)$ Σ的基网
2. D个体集（D上的运算集为Ω）
3. V为D上的变量集
4. $A_P:P\to\pi$（可变谓词集） π为D上的可变谓词集 $A_P(p)$是n元谓词
5. $A_T:T\to f_D$（D的公式集） $A_T(t)$是静态谓词加Ω上的运算符
6. $A_F:F\to f_S$（D的符号和集）
   $A_F(t,p)$或$A_F(p,t)$n元符号和且$A_T(t)$与$A_F(t,p)$, $A_F(p,t)$的自由变量相同
   有向弧上的标记用符号和表示，为了便于判断变量替换是否可行。
7. $M_0:P\to f_s$ $M_0(p)$是n元符号和
   作为系统状态的标识可以看成是为每个谓词指明其等同的个体子集。为了便于给出变迁规则，定义中用符号和来表示子集。

- **变迁规则**

  设M为Pr/T_-系统$\sum$的一个标识

  - 变迁$t\in T$在M有发生 权的充分必要条件是：存在M下的可行替换$t(z_1\leftarrow d_1,z_2\leftarrow d_2...,z_l\leftarrow d_l)$,其中$\{z_1,z_2,...\}$,为与$t$有关的符号和及公式中的自由变量集。$M$和$t$的这种发生权记作：

    $M[t(z_1\leftarrow d_1,z_2\leftarrow d_2,..z_l\leftarrow d_l)>$
  - 若$M[t(z_1\leftarrow d_1,z_2\leftarrow d_2,...,z_l\leftarrow d_l)>$,则$t$可按可行替换$t(z_1\leftarrow d_1,z_2\leftarrow d_2,..,z_l\leftarrow d_l)$发生，其后继标识$M^{'}$为：对所有$p\in P$,

    $M^{'}(p)=M(p)-A_F(p,t)(z_1\leftarrow d_1,z_2\leftarrow d_2,...z_l\leftarrow d_l)+A_F(t,p)(z_1\leftarrow d_1,z_2\leftarrow d_2,...,z_l\leftarrow d_l)$

- **有色网的定义**

  $\sum=(P,T;F,C\ ,\ I_-,I_+,M_0)$称为有色网的充分必要条件

  - $(P,T;F)$为有向网，称为$\sum$的基网
  - $C:P\cup T\to N(D),N(D)$为颜色集D之幂集合
  - 对$p\in P,C(P)$是库所p上所有可能的token色集合
  - 对$t\in T,C(t)$是变迁t上所有可能的出现色集合
  - I_-,I_+ 分别是PXT上的负函数和正函数，使得对所有$(p,t)\in PXT$
    - $I_-(p,t)\in[C(p)_{MS}\to C(t)_{MS}]_L$,且$I_-(p,t)=0$充分必要条件是$(p,t)\notin F$
    - $I_+(p,t)\in[C(t)_{MS}\to C(p)_{MS}]_L$,且$I_+(p,t)=0$充分必要条件是$(t,p)\notin F$
  - $M_0:P\to D_{MS}$,称为$\sum$的初始标识，它必须满足条件$\forall p\in P:M_0(p)\in C(p)_{MS}$

    即$M_0(p)$是p的token色集合上的多重集

- **有色网的变迁规则**

  设X为有色网$\sum$在标识M下的一步，则X发生后的后继标识$M^{'}$,由下式给出：

  $\forall p\in P,M^{'}(p)=M(p)+\sum_{t\in T}I_+(p,t)(X(t))-\sum_{t\in T}I_-(p,t)(X(t))$

第六章 条件/事件系统

- **完全情态集的定义**

基本网系统$(B,E;F,C_{in})$的完全情态集是满足下列条件的最小集合$[C_{in}]$

- $c_{in} \in [c_{in}]$
- 若$c_1 \in [c_{in}], c_2 \subseteq B$, 且有$u \subseteq E$,使得$c_1[u > c_2$或者$c_2 < u]c_1$,则$c_2 \in [c_{in}]$

- **情态集**

EN_-系统$N=(B,E;F,c_{in})$的情态集C_N是满足下列条件的最小集合

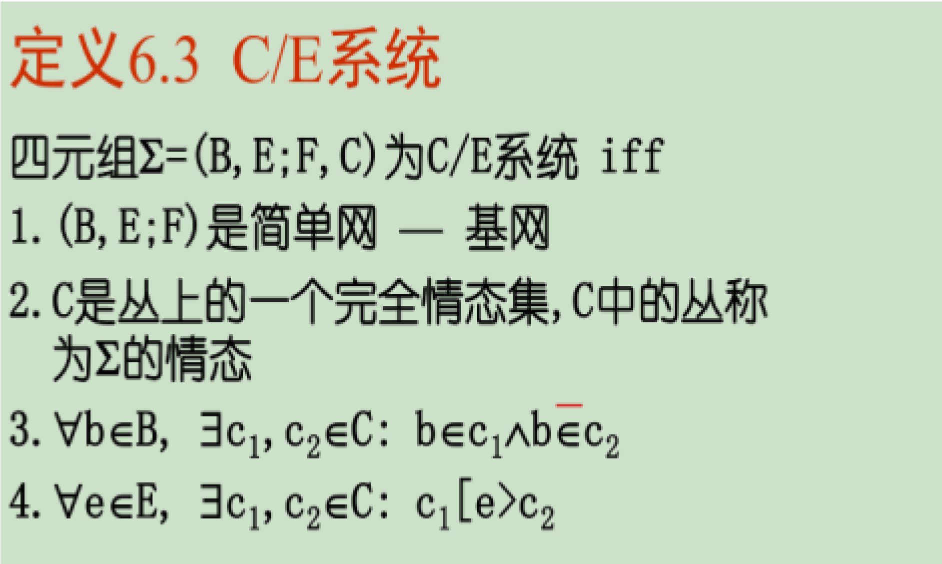
$c_{in} \in C_N$

若$c \in C_N, u \subseteq E, c^{'} \subseteq B$且 c[u > c^{'}, 则 $c^{'} \in C_N$

- **完全可达关系是丛上的等价类**

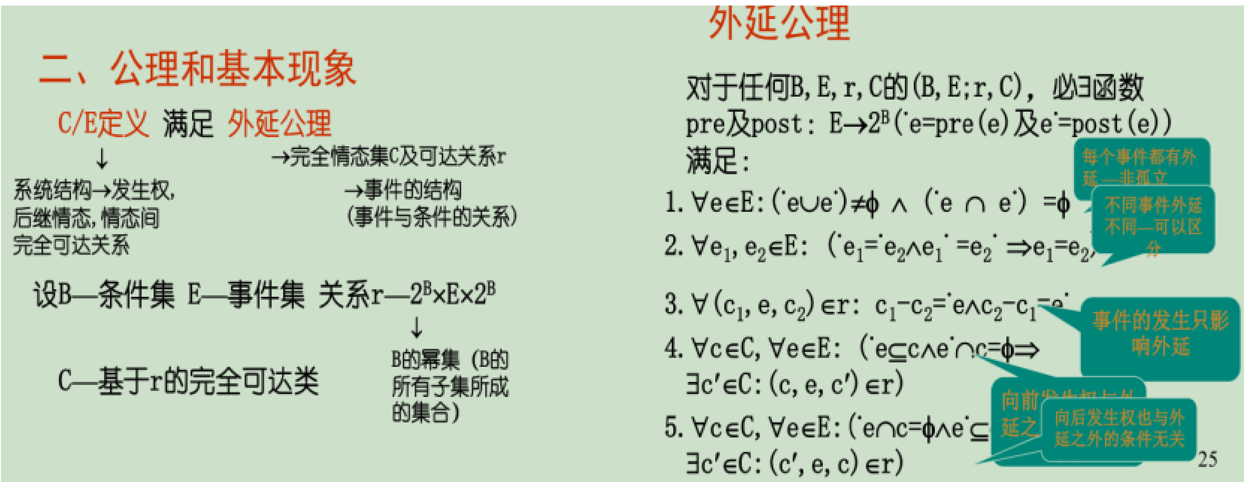
$(B,E;F,c_{in})$是EN系统，设$c_1, c_2 \subseteq B$, 则$[c_1]=[c_2]$或者$[c_1] \cap [c_2] = \emptyset$

- **C/E系统的定义**

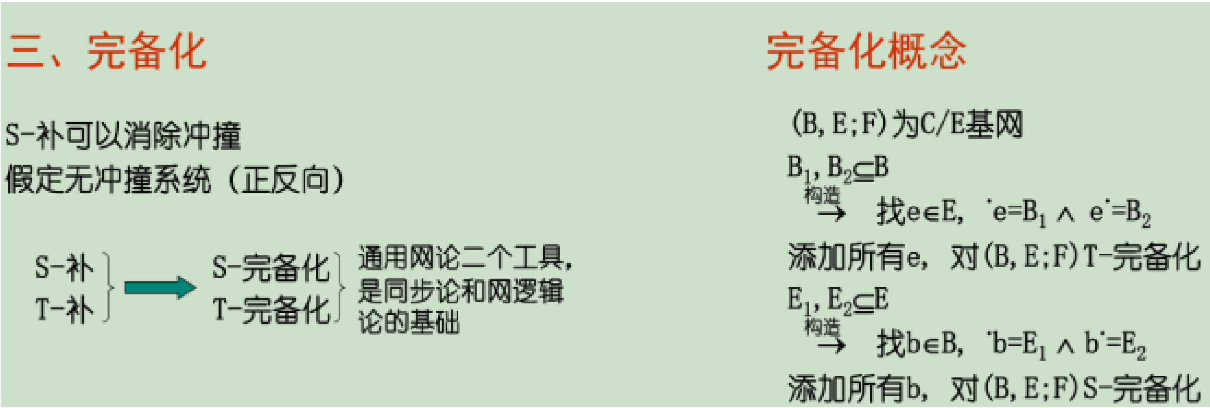


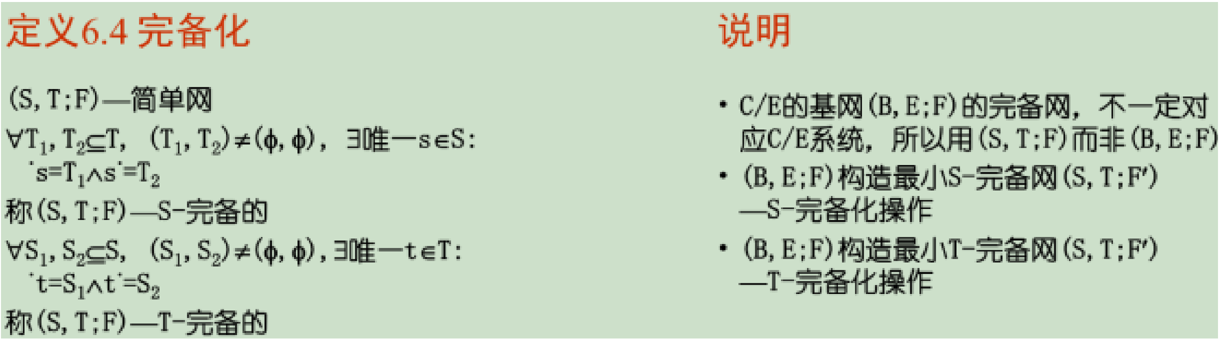
- **如何理解C/E系统没有初始标识**

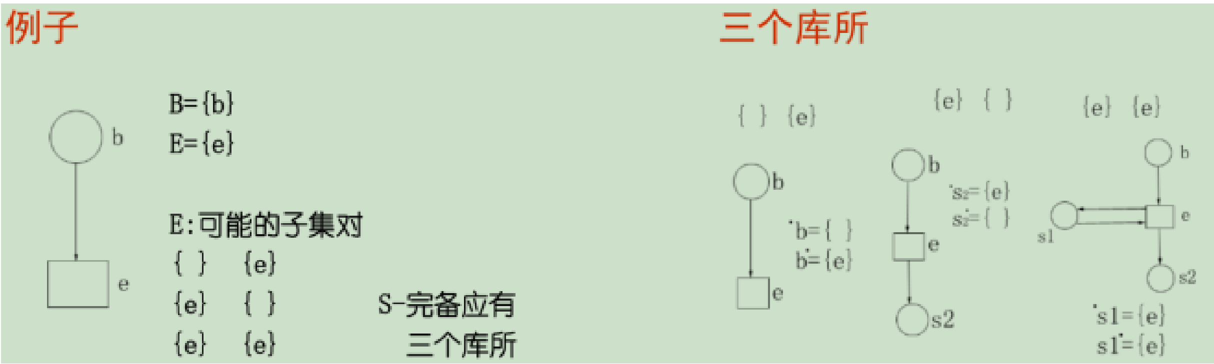
- **外延公理**



- **S-完备T-完备**







第七章 同步论

- **同步距离的定义**

  令$E_1, E_2 \subseteq E$为$C/E_$系统$\sum = (B, E; F, C)$的任意两个非空事件子集，则E_1和 E_2的同步距离$\sigma(E_1, E_2)$由下面公式给出

  若有最大值 $\sigma(E_1, E_2) = \max_{p\in\pi}\{Occ(E_1, p) - Occ(E_2, p)\}$

  若无最大值$\sigma(E_1, E_2) = \infty$

  p为进程

- **同步距离的性质**

定理7.1 同步距离性质

C/E系统中, $a, b, c, d \subseteq E$为事件集, 则

1. $\sigma(a, b) \geqslant 0$
2. $\sigma(a, b)=0 \Leftrightarrow a=b$
3. $\sigma(a, b)=\sigma(b, a)$
4. $\sigma(a, b)+\sigma(b, c) \geqslant \sigma(a, c)$
5. $\sigma(a, b)=\sigma(a-b, b-a)$
6. $\sigma(a \cup b, c \cup d) \leqslant \sigma(a, c)+\sigma(b, d) + \sigma(a \cap b, c \cap d)$

- **同步距离与系统行为**

二、同步距离与系统行为

$E_1, E_2 \subseteq E$

$\sigma(E_1, E_2)=\infty \Rightarrow E_1$中事件可以比$E_2$中事件多(或少)发生任意次 $\Rightarrow$ 异步

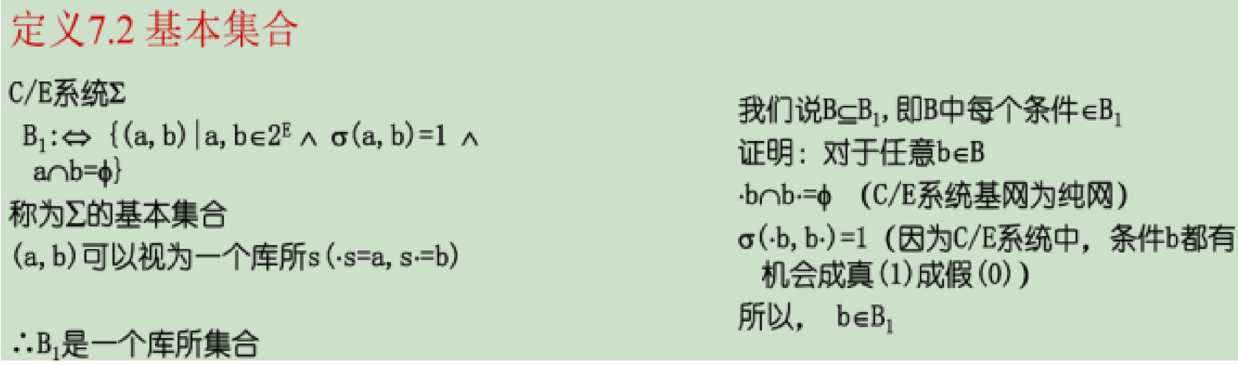
$\sigma(E_1, E_2)<\infty \Rightarrow E_1$中事件最多比$E_2$中事件多(或少)发生$\sigma(E_1, E_2)$次

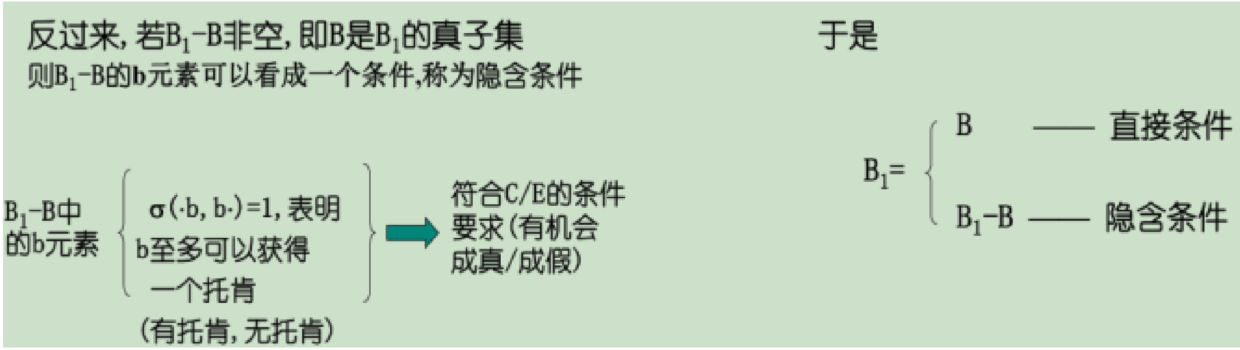
由同步距离性质，我们已知

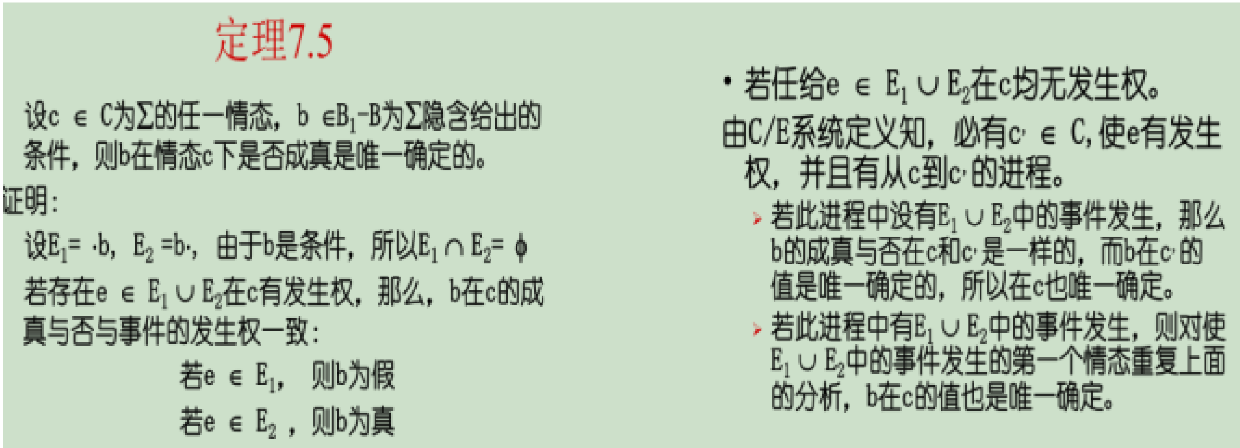
$\sigma(E_1, E_2)=0 \Rightarrow E_1=E_2$

$\sigma(E_1, E_2)=1 \Rightarrow E_1, E_2$中事件只能交替发生。同步距离为1的两组事件对应着系统中的一个条件，这些条件对研究系统性质有重要意义。

- **基本集合B_1和条件集B的关系**







- **计算同步距离的方法**

第八章 同步论

- ## 并发的定义：

  设X为出现网中包含所有元素的集合，元素之间的因果关系及由因果关系产生的顺序用“<”

  严格偏序集(X,<)：

  $$(<\subseteq X\times X)\wedge(<\cap Id=\Phi)\wedge(<^{2}\subseteq<)$$

  对严格偏序集可以定义无序关系。并发关系就是出现网元素之间由<产生的无序关系。

  定义 $x\ co\ y \Leftrightarrow x,y\in X\wedge\neg(x<y)\wedge\neg(y<x)$

  即： $co=\overline{<\cup>}$

并发关系是否有传递性？若有就是一个等价关系，从而把X分解为等价类，等价类内的元素并发，等价类由(X, >)确定了完全的顺序。如果出现网是一个信号留下的轨迹，X退化为一条线，并发才可能是传递的。系统进程一般不会只有一个信号，几个信号也不会互不相关，因此，一般而言，并发是不传递的。

令 $Co(x)=\{y\,|\,(x,y)\in co\}$，若x不等于y，但是Co(x)=Co(y),那么，对关系并发而言，x和y虽然不同确是无法区分的。我们把x和y的这种关系记为 $\tilde{co}$

$x\ \tilde{co}\ y:\Leftrightarrow Co(x)=Co(y)$ 称 为co的传递核。

在研究并发时我们不关心不能用并发关系区分的元素。即要求x $\tilde{co}$ y→x=y。