博弈论

上课时间：每周一晚上18:10-19:50

上课地点：玉泉路校区 教学楼阶一1

授课团队：兴军亮（教师）、徐航（助教）

上一讲内容回顾

- 信息概念、静态特点
- 博弈定义与表示
- 均衡分析与求解
- 均衡的存在性

知识点

能力线

- **完全信息静态博弈**：常见博弈均衡的求解方法
- **不同类型均衡结果**：均衡存在性的判定方法

- **完全信息静态博弈**：实际生活中如何决策？

① 全面考虑所有信息

② 勿被虚假信息迷惑

价值面

2023春季学期 · 本科生专业课 · 《博弈论》

第三讲：完全信息动态博弈

兴军亮

授课时间：2024年3月11日

联系方式：jlxing@nlpr.ia.ac.cn

第三讲 完全信息动态博弈

内容提纲

1. 博弈类型定义及表示
2. 扩展式博弈均衡分析
3. 子博弈精炼纳什均衡
4. 扩展式博弈示例讲解

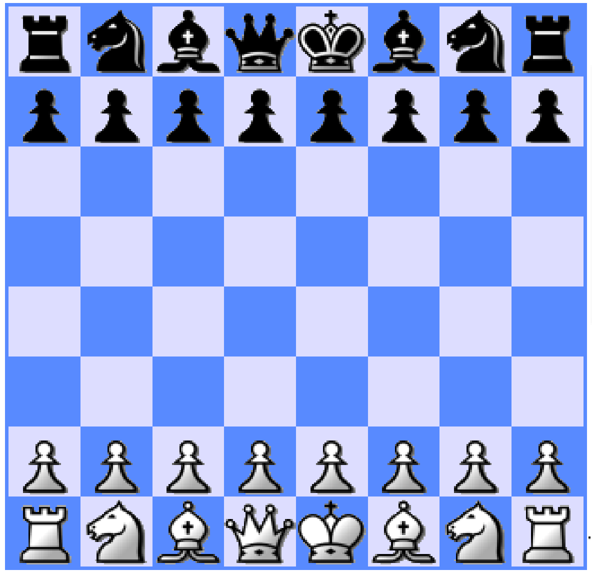
第三讲 完全信息动态博弈

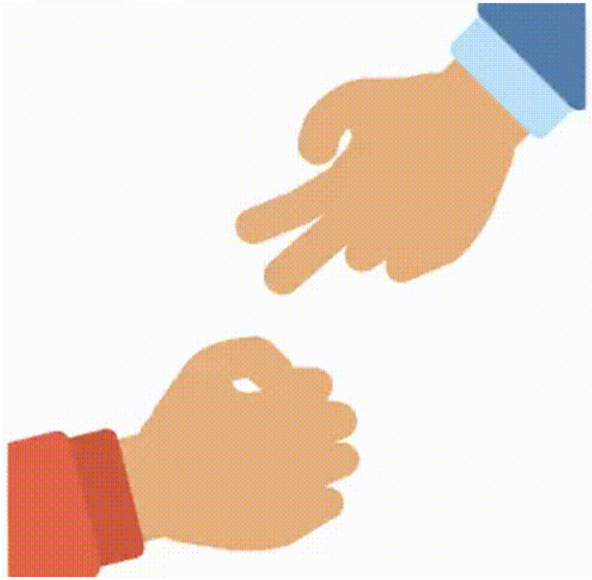
内容提纲

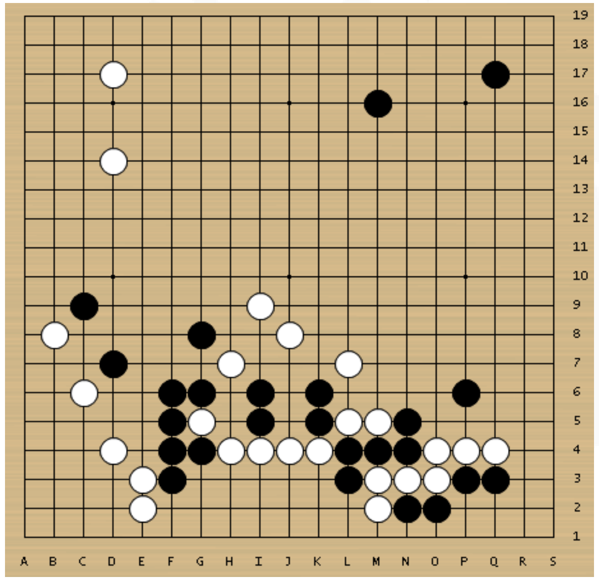
1. 博弈类型定义及表示
2. 扩展式博弈均衡分析
3. 子博弈精炼纳什均衡
4. 扩展式博弈示例讲解

完全信息动态博弈

- 现实世界中博弈的很多实际应用问题中都存在着多阶段决策的动态结构，为了表示和分析这种动态结构，博弈研究者提出了动态博弈这种类型的博弈加以研究。
- 当动态过程中的博弈各方完全知道博弈的信息，就是我们这一讲要的完全信息动态博弈。







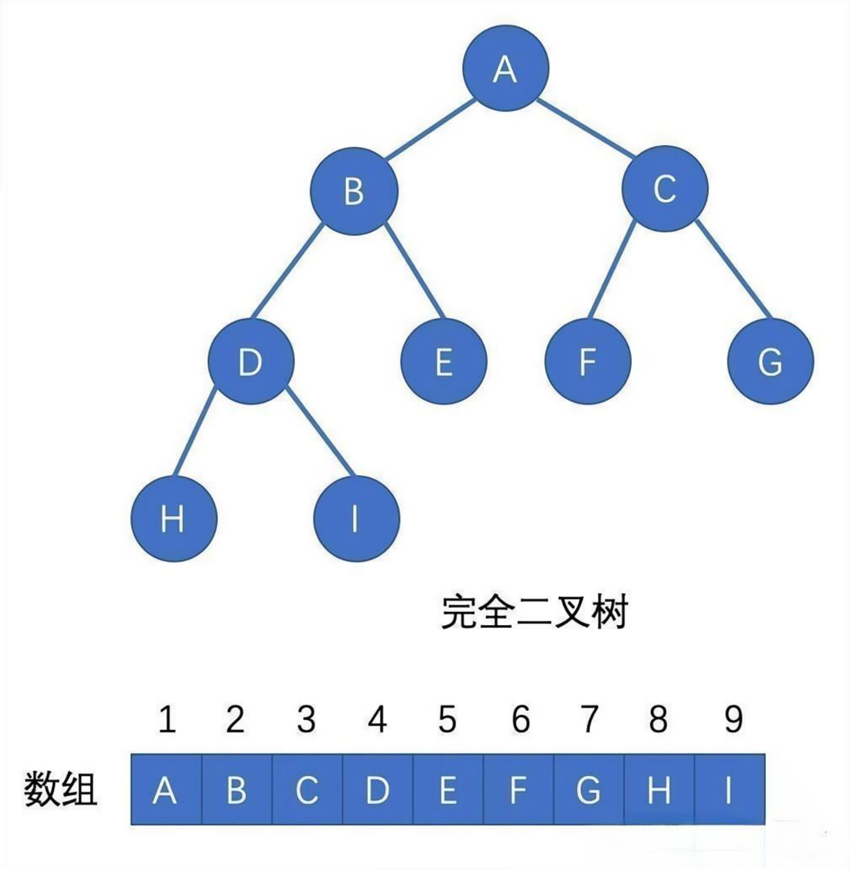
完全信息动态博弈

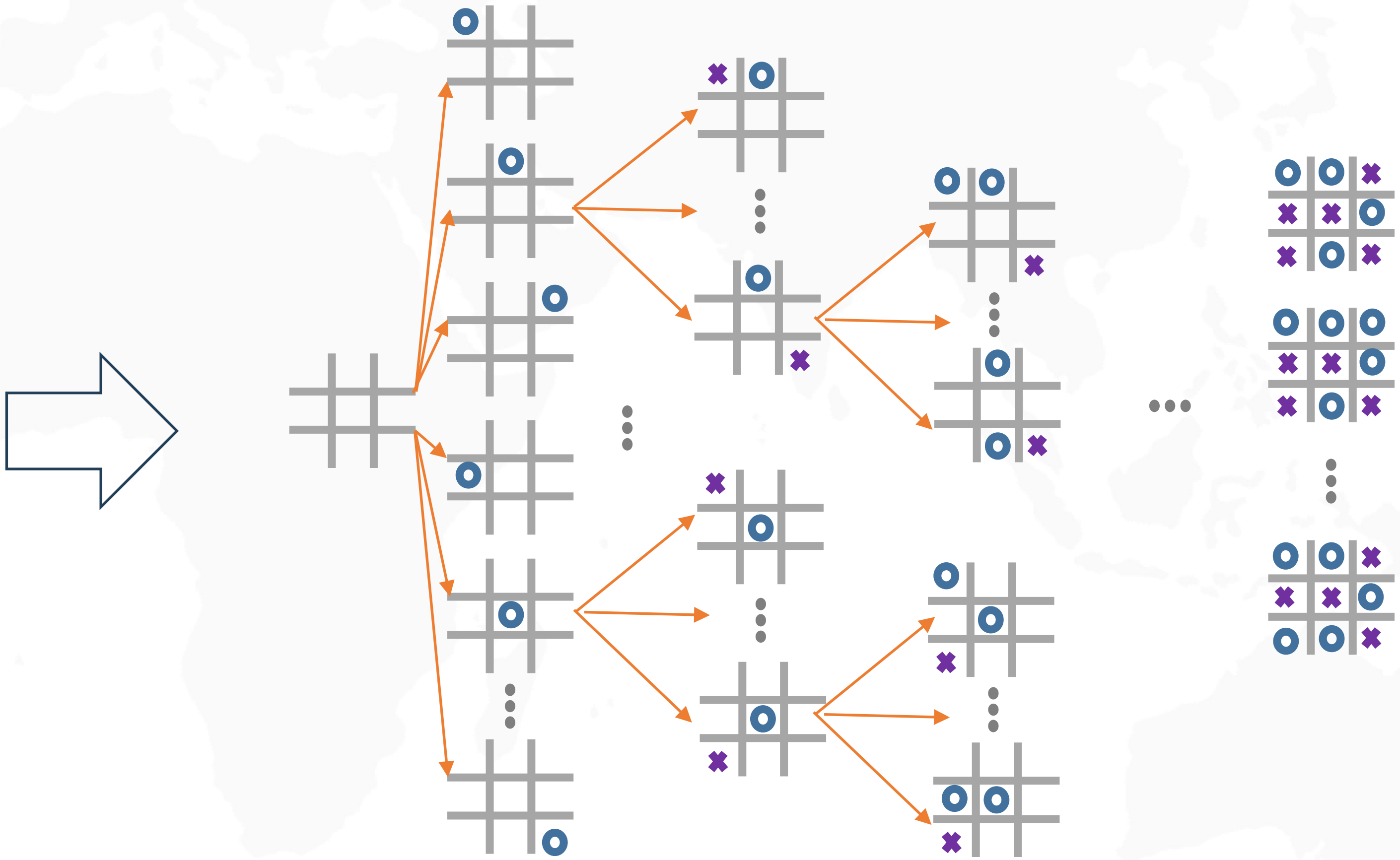
- 博弈定义：博弈过程中参与人预先知道关于博弈的所有信息（完全信息），参与人的行动有先后顺序（动态博弈）的博弈。
- 本讲主要关注后行动者能够观察到先行动者行动的完全信息动态博弈（完美信息，不完美信息博弈在第五讲中讲解，与不完全信博弈相互可以转换）。
- 完全信息动态博弈需要反映出来的要素

① 参与人集合：$i \in N, N = (1,2,\dots,n)$，参与人中可以包括自然
② 参与人的行动顺序：谁在什么时候行动
③ 参与人的行动空间：每次行动时，有什么选择
④ 信息集合：每次行动结束，参与人知道些什么
⑤ 效益函数：博弈结束后，参与人知道将得到什么
⑥ 外生事件（即自然选择）的概率分布

博弈的展开式表示

- 博弈树：采用类似于数据结构中的树结构表示有限动态博弈





如何对博弈树中出现的对象进行刻画和定义？

博弈树中的重要概念

- **结点**：X表示所有结点的集合，$x \in X$表示某个特定结点，具体可分为决策结点（中间结点，包括初始结点、机会结点和参与人决策结点）和终止节点（叶结点）；我们用关系符号$\prec$表示X上结点的先后顺序关系，$x^1 \prec x^2$表示x^1在x^2之前，x^1是x^2的父辈结点，x^2是x^1的子孙结点，关系$\prec$满足传递性（$x^1 \prec x^2, x^2 \prec x^3 \Rightarrow x^1 \prec x^3$）和反对称性（$x^1 \prec x^2 \Rightarrow \dashv x^2 \prec x^1$），表明关系$\prec$是偏序的，即某些结点之间不存在先后关系；我们使用$P(x)$表示$x$之前所有结点的集合（父辈结点集合），简称为$x$的前列集，使用$T(x)$表示$x$之后所有结点的集合（子孙结点集合），简称为$x$的后续集，初始结点的前列集为空，叶子结点的后续集也为空，某个结点的前列集是全序的，即集合中每两个结点间都存在先后顺序关系；对于$\forall\, x \in X$，如果存在$p(x) \in P(x)$使得$\forall x' \prec x, x' \prec p(x)$，$p(x)$就称为$x$的直接前列结，类似地可定义$t(x)$为$x$的直接后续结。在没有歧义的情况下，$p(x)$和$t(x)$也可代表相应结点的集合。

博弈树中的重要概念

- **树枝**：博弈树中从一个决策结点到其直接后续结的连线，树枝代表一个可供树枝起始节点所对应的参与人选择的行动（动作），以一个决策结点为起始结点的所有树枝代表了所有可供树枝起始节点所对应参与人选择的行动（动作）集合；从结点x到其一个直接后续结x'的树枝可表示为xx'或$x \to x'$。
- **路径**：博弈树中如果两个结点x^1和x^2之间存在顺序关系，比如$x^1 \prec x^2$，那么我们就说从结点x^1到结点x^2可达，结点x^1到结点x^2经过的所有结点序列（包括x^1和x^2）称为结点x^1到结点x^2的<u>路径</u>，路径上所有结点的集和称为结点x^1到结点x^2的<u>路径集</u>。路径不要求始于初始结点和终于终止结点。
- **信息集**：博弈决策结点集的一个子集，是某个参与人无法区分的决策节点组成的集合，满足：1）每个决策结点都是同一参与人的决策结点；2）该参与人知道博弈进入该节点子集，但不知道自己具体在哪一个决策节点。通常使用虚线或者椭圆将博弈树中属于同一信息集的结点连接起来。

博弈树中的重要概念

- **结点参与人函数**：用于表示在某个结点轮到谁行动；引入函数$i: X \to \{1, \cdots, N\}$，$i(x)$定义为从决策结点集合到参与人集合（包括自然）的映射，解释为在决策结x该由参与人i做出行动。
- **结点的行动集合**：用于表示结点可选择的候选动作集合。我们使用$A(x)$表示结点x参与人$i(x)$的候选动作集合（行动集合）。
- **结点所在信息集**：表示结点x所在信息集的结点信息。我们使用$h(x)$表示结点x所在信息集所包含的所有结点集合 。
- **参与人的动作集**：用于表示参与人的所有候选动作集合。我们用$A(i)$或A_i表示参与人i的候选动作集合。
- **参与人的信息集**：用于表示参与人的所有信息集。我们用$\mathbb{I}(i)$或$\mathbb{I}_i$表示参与人i的所有信息集。
- **博弈的动作集**：博弈所有参与人包括自然的动作集合，用A表示，$A \triangleq \cup_{i=1}^{N} A(i)$。
- **博弈的信息集**：博弈的信息集集合，用$\mathbb{I}$表示，$\mathbb{I} \triangleq \cup_{i=1}^{N} \mathbb{I}_i$ 。

博弈树中的重要概念

- **历史**：与路径比较接近的一个概念，路径对起始和终止结点没有要求，历史默认要求从起始节点开始，完整历史（终止历史，terminal history）更要求到叶子结点结束；我们使用$\mathbb{H}$表示由所有终止历史组成的集合，其中一个终止历史是一条从根节点到终止节点的行动路径；以$S_{\mathbb{H}}$表示由所有终止历史的所有真子历史（包括空历史ϵ）组成的集合，目的是为了表示从初始结点到所有决策结点的历史，可以称为非终止历史集合。同样可以在非终止历史集合的基础上定义参与人函数，我们用$P: S_{\mathbb{H}} \rightarrow \{1, \dots, N\}$来表示一个参与人函数，它将每个真子历史与相应的参与人联系在一起，$P(h)$表示对于历史h轮到参与人$P(h)$进行行动，其中$h \in S_{\mathbb{H}}$；在博弈学习研究方向，历史也通常与episodes、trajectories等概念关联。

博弈的展开式表示举例

- 可观察情形下的硬币配对：有两个参与人（1和2），每人有一个硬币，其中一个人将自己的硬币放下，正面朝上（以H表示）或反面朝上（以T表示），另一人**看到这个结果**，然后也将自己的硬币放下。如果两枚硬币配对成功（都正面朝上或者反面朝上），那么参与人2给1一元钱，反之参与人1给2一元钱。图1和图2是该博弈的两个版本，分别为参与人1先放硬币和参与人2先放硬币。

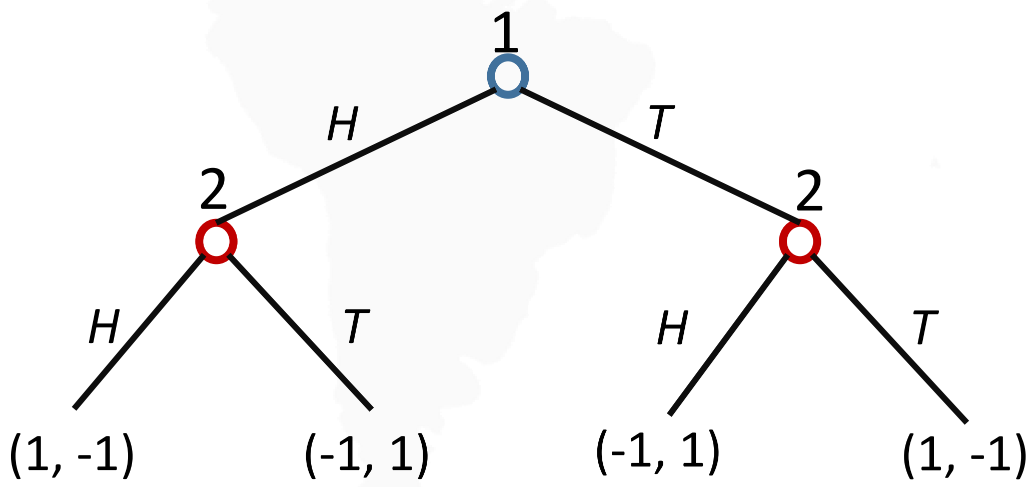


图1：可观察情形下的硬币配对（参与人1先行动）

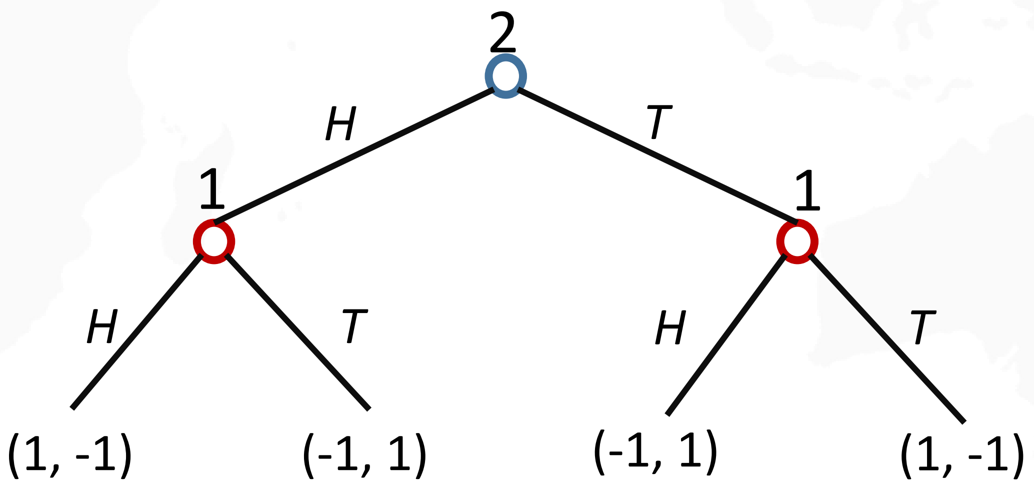


图2：可观察情形下的硬币配对（参与人2先行动）

博弈的展开式表示举例

- 不可观察情形下的硬币配对：有两个参与人（1和2），每人有一个硬币，其中一人将自己的硬币放下，正面朝上（以*H*表示）或反面朝上（以*T*表示），另一人**看不到这个结果**，然后也将自己的硬币放下。如果两枚硬币配对成功（都正面朝上或者反面朝上），那么参与人2给1一元钱，反之参与人1给2一元钱。图1和图2是该博弈的两个版本，分别为参与人1先放硬币和参与人2先放硬币。

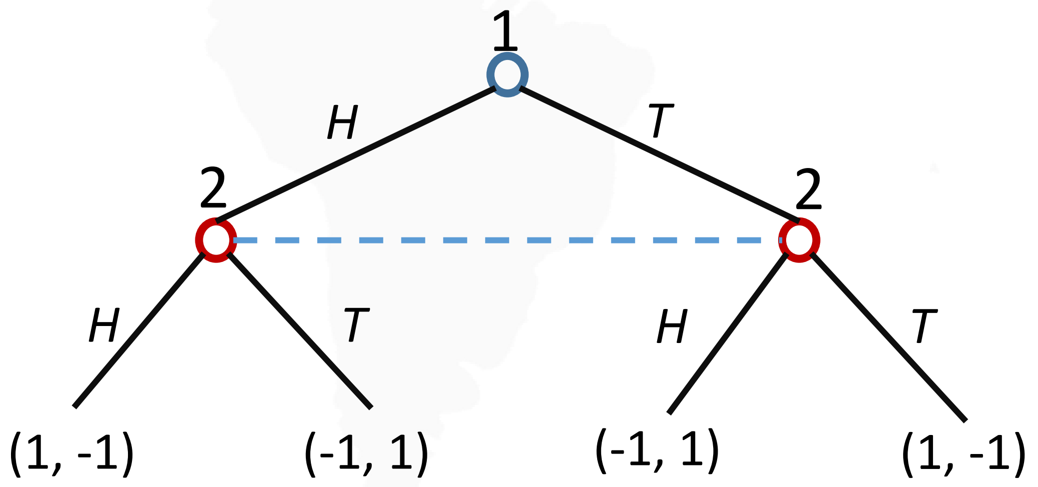


图1：不可观察情形下的硬币配对（参与人1先行动）

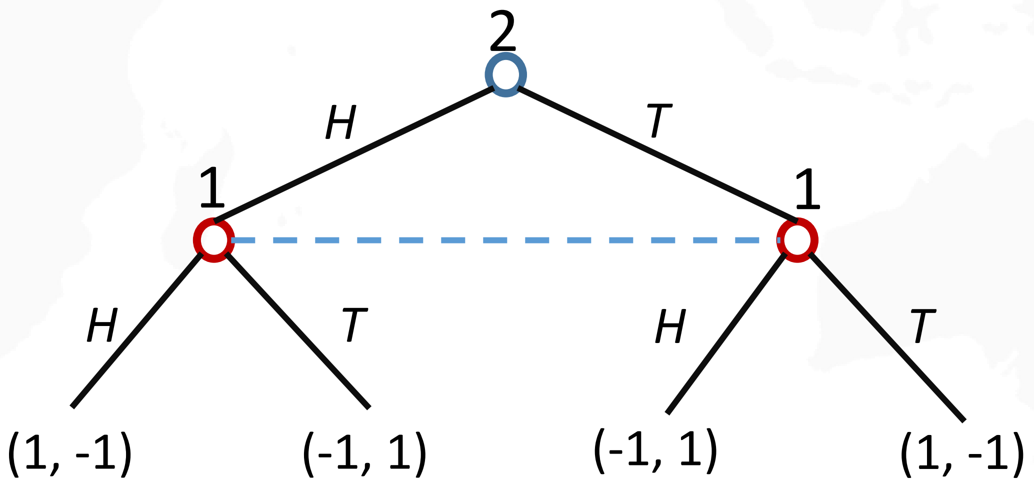


图2：不可观察情形下的硬币配对（参与人2先行动）

博弈的展开式表示举例

- 房地产开发博弈中的不同信息集
  - 图1：B知道A和N的选择，共7个信息集
  - 图2：B知道A但不知N的选择，共5个信息集
  - 图3：B知道N但不知A的选择，共5个信息集

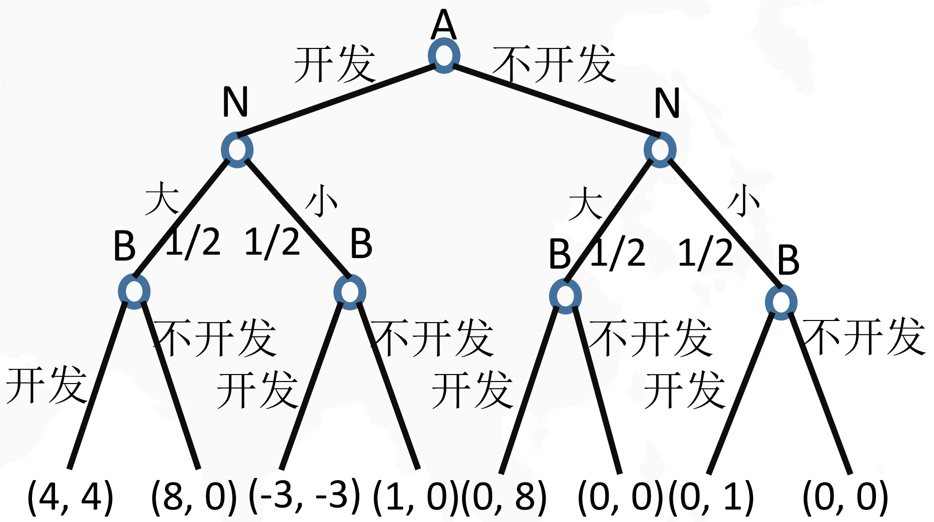


图1：房地产开发博弈（完全信息）

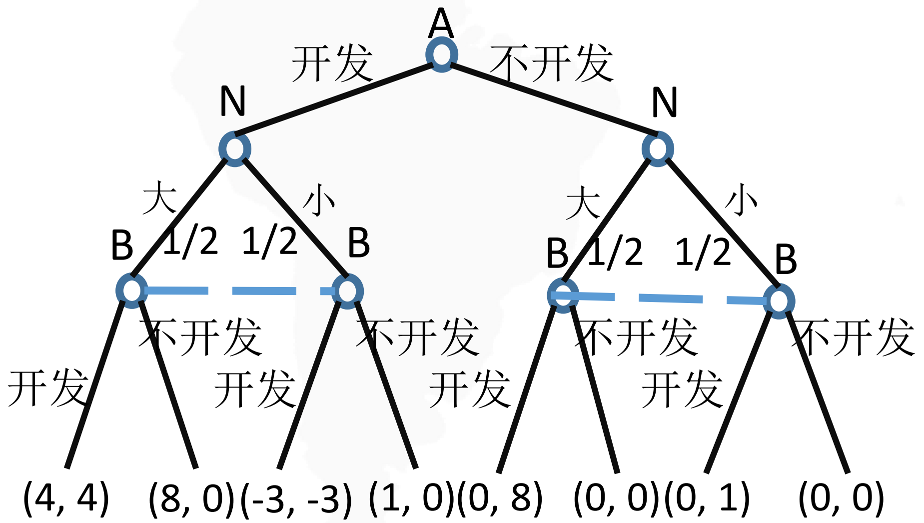


图2：房地产开发博弈（B知道A但不知道N）

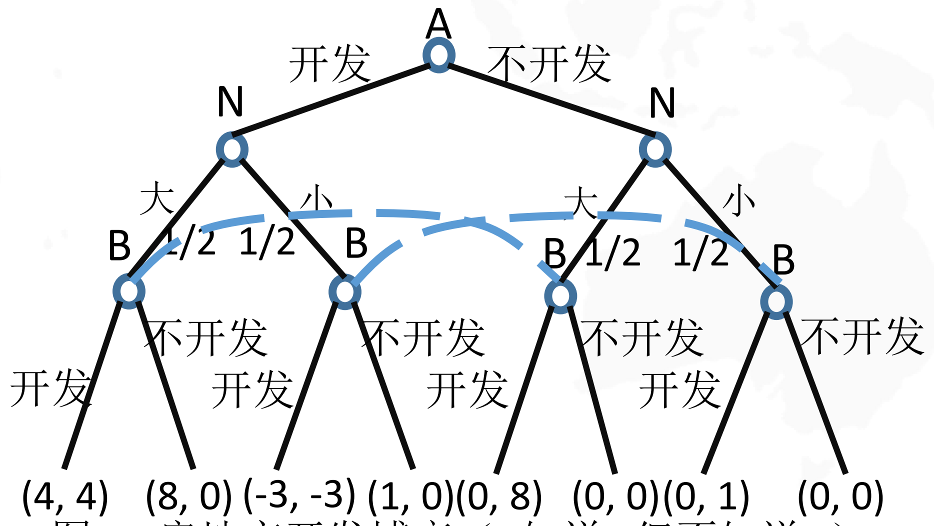


图3：房地产开发博弈（B知道N但不知道A）

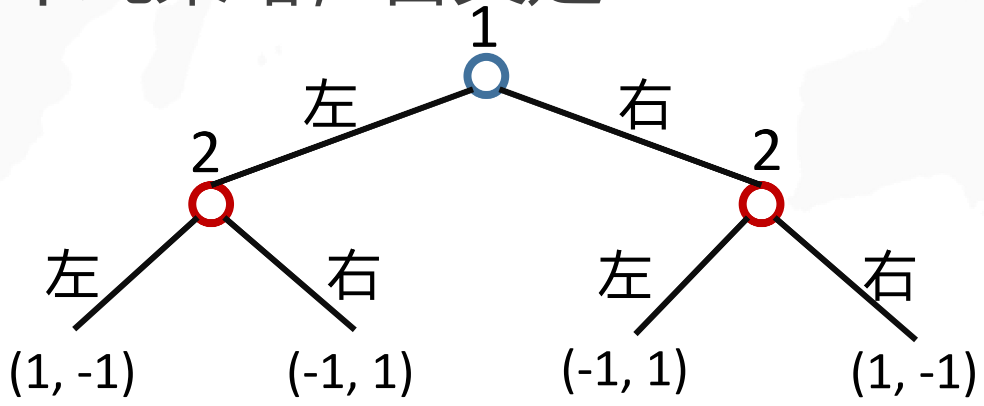
扩展式博弈的纯策略集合

- 参与人i在其所有决策点（信息集）的动作选择

扩展式博弈中的纯策略

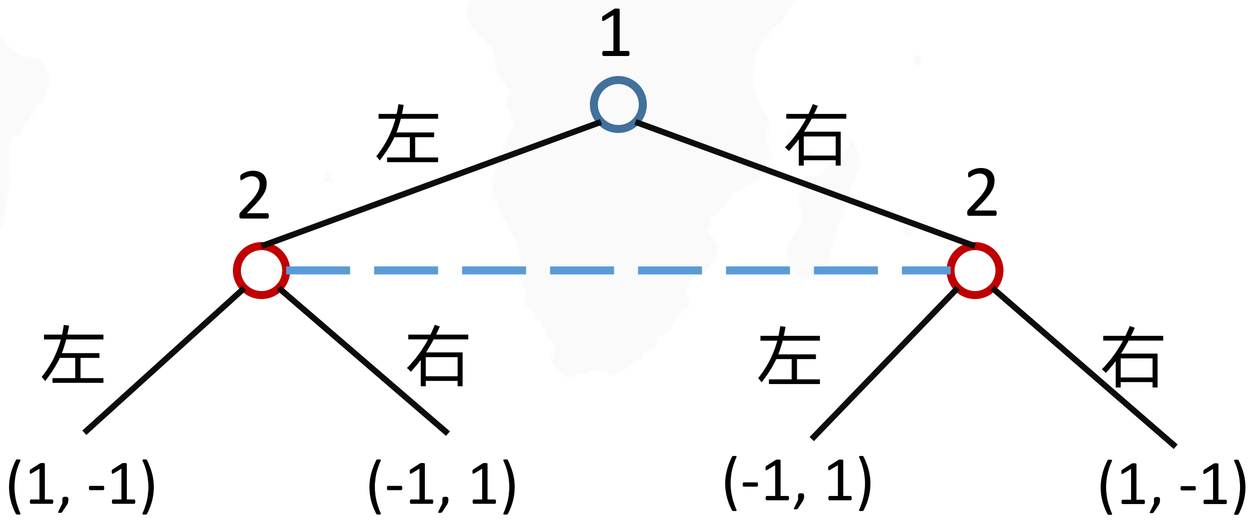
令$A(h)$表示在信息集h下的行动集，$H_i \equiv \{h_i\}$是参与人i的信息集集合，$A_i \equiv \cup_{h_i \in H_i} A(h_i)$为参与人$i$所有可选择行动的集合，参与人$i$的纯策略集合$S_i \equiv \times_{h_i \in H_i} A(h_i)$。

- 每一个参与人i的纯策略集合S_i：
  - $S_1 = \{$左，右$\}$，玩家1有两个纯策略
  - 玩家2有两个决策点，每个决策点2种动作选择→4个纯策略，$S_2 = \{$左左，左右，右左，右右$\}$，玩家2有4个纯策略，含义是
    - 左左：不管玩家1如何选择玩家2都选左
    - 左右：1左2也左，1右2也右
    - 右左：1左2右，1右2左
    - 右右：不管1如何，2都右



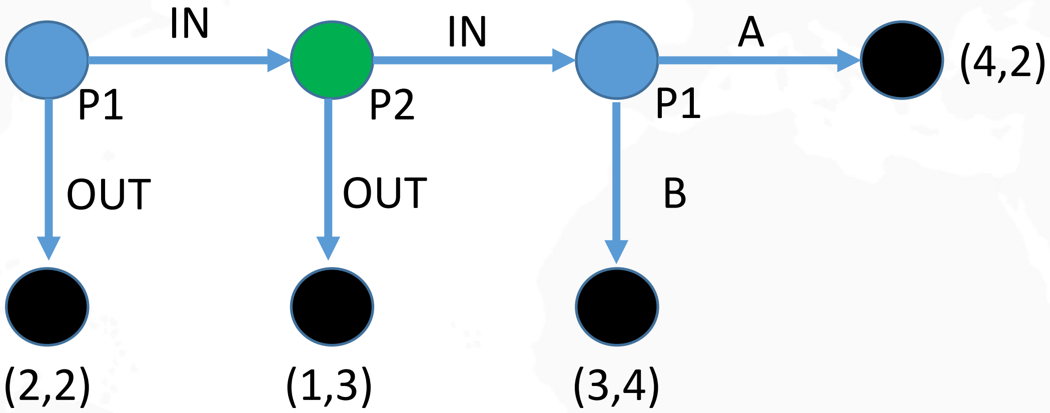
扩展式博弈的纯策略集合

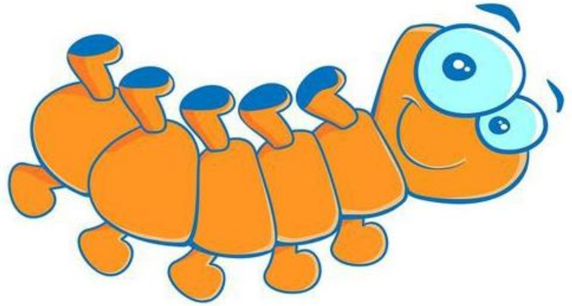
- 不完美信息条件下，每个参与人i的纯策略集合S_i：
  - $S_1 = \{$左，右$\}$，玩家1有2个纯策略
  - $S_2 = \{$左，右$\}$，玩家2也有2个纯策略
  - 玩家2为什么只有2个纯策略？和上个例子有什么不同？
  - 因为玩家2只有1个决策点（信息集），该信息集下只有2个动作可供选择，因此玩家2只有两个纯策略



扩展式博弈的纯策略集合

- 蜈蚣博弈（The Centipede Game）

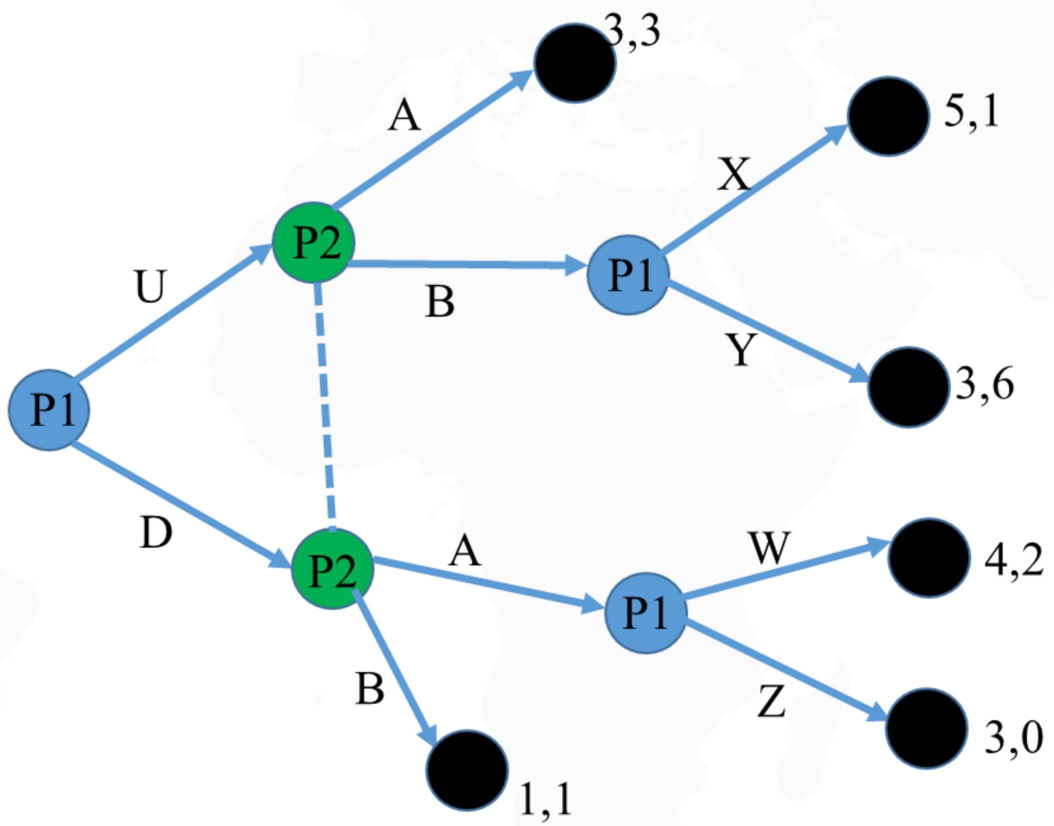




- S_2 ={IN, OUT}
- S_1 ={IN A, IN B, OUT A, OUT B}
  - 很多人会有疑问：P1选了OUT之后博弈就结束了，为什么还要指定后续的动作？
  - 回忆定义：扩展式博弈的一个纯策略描述了参与者在其所有决策点（信息集）的动作选择，不能只指定部分决策点的选择！

扩展式博弈的纯策略集合

- 不完美信息博弈



S_1 ={UXW, UXZ, UYW, UYZ, DXW, DXZ, DYW, DYZ }

S_2 ={A,B}，玩家2只有1个决策点！

扩展式博弈的混合策略

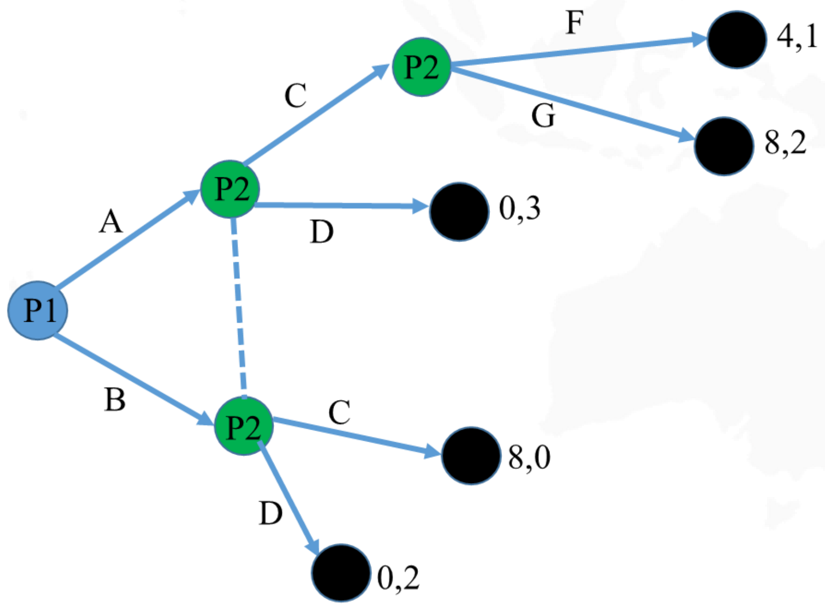
- 与矩阵式博弈相同：

混合策略（Mixed Strategy）

假设i有K个纯策略：$S_i=\{s_{i1},\ldots,s_{iK}\}$，概率分布$\sigma_i=(\sigma_{i1},\ldots,\sigma_{iK})$为$i$的一个混合策略，$\sigma_{ik}$是$i$选择纯策略$s_{ik}$的概率。

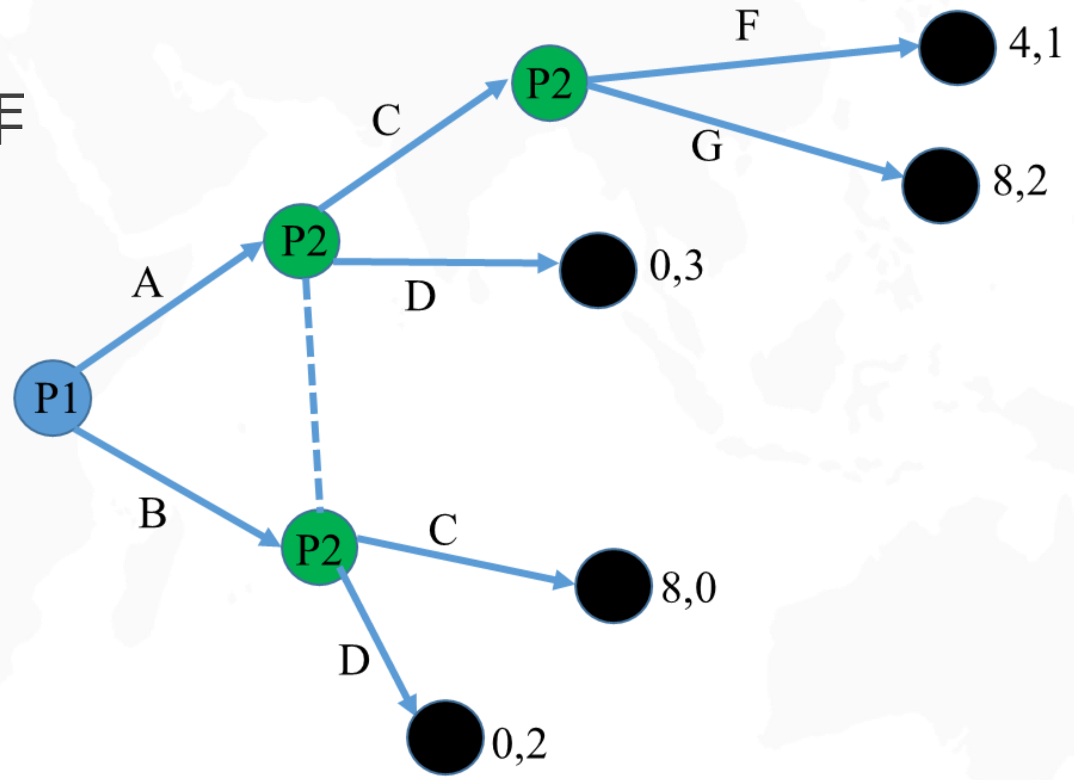
玩家1的纯策略集合S_1 ={A,B}，它的一个混合策略σ_1为S_1上的某一概率分布

玩家2的纯策略集合S_2 ={CF,CG,DF,DG}，它的一个混合策略σ_2同样为S_2上的某一概率分布



扩展式博弈的行为策略

- 玩家i的行为策略（Behavioral Strategy）：
  - 为玩家i每一个决策点指定一个概率分布
  - 玩家1的行为策略：
    - α的概率选A动作，以$1-\alpha$的概率选B动作
  - 玩家2的行为策略：
    - 第一个决策点，β概率选C，$1-\beta$概率选D，第二个决策点，γ概率选F，$1-\gamma$概率选G

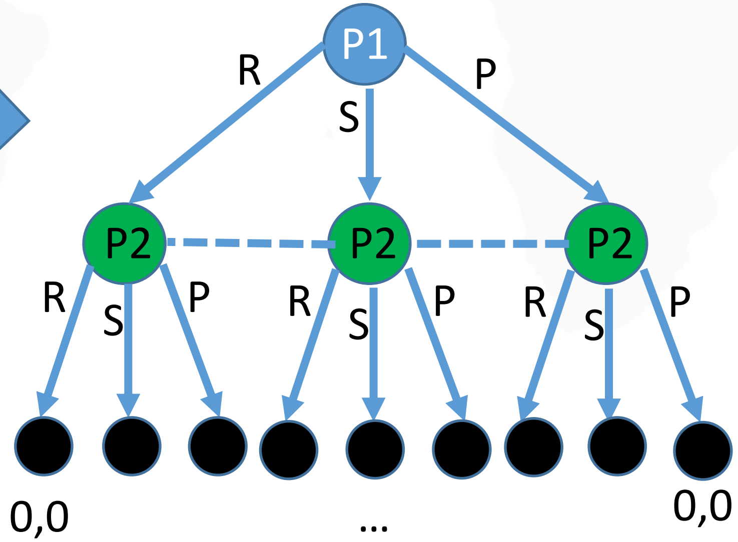


矩阵式表示和扩展式表示的转换

- 博弈的矩阵式表示和扩展式表示可以互相转化
- 矩阵式转化为扩展式
  - 利用信息集来表示动作同时进行

玩家1 \ 玩家2	石头	剪刀	布
石头	0, 0	1,-1	-1,1
剪刀	-1,1	0,0	1,-1
布	1,-1	-1,1	0,0

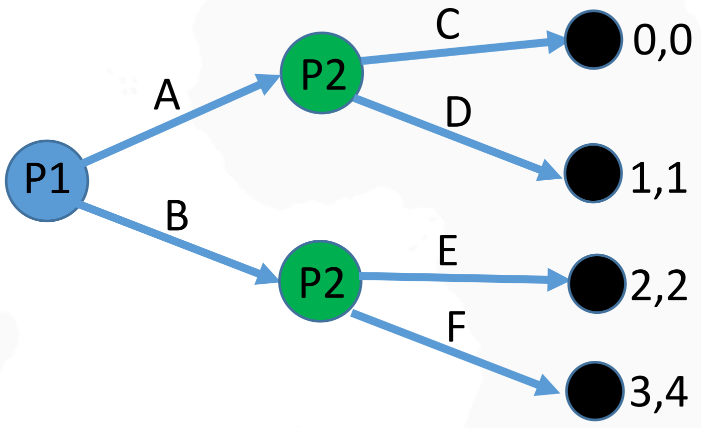
石头剪刀布的
矩阵式表示



石头剪刀布的
扩展式表示

矩阵式表示和扩展式表示的转换

- 扩展式转化为矩阵式
  - 首先写出所有玩家的纯策略集合，集合大小确定了矩阵每一维度的大小，然后用收益函数进行填空



扩展式表示

S_1 ={A,B}

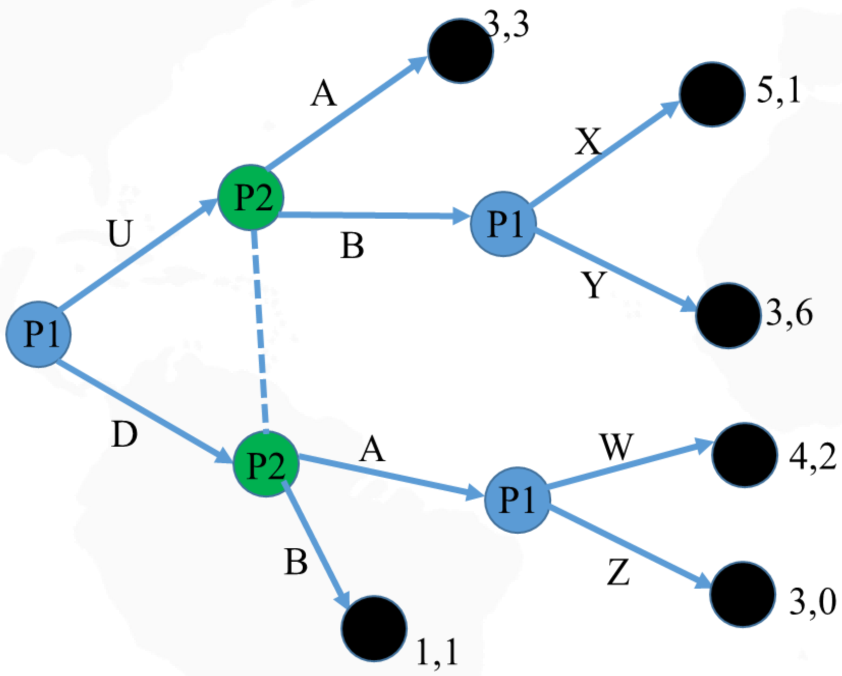
S_2 ={CE, CF, DE, DF}

	CE	CF	DE	DF
A	0,0	0,0	1,1	1,1
B	2,2	3,4	2,2	3,4

矩阵式表示

矩阵式表示和扩展式表示的转换

- 扩展式转化为矩阵式



扩展式表示

S_1 ={UXW, UXZ, UYW, UYZ, DXW, DXZ, DYW, DYZ }

S_2 ={A, B}

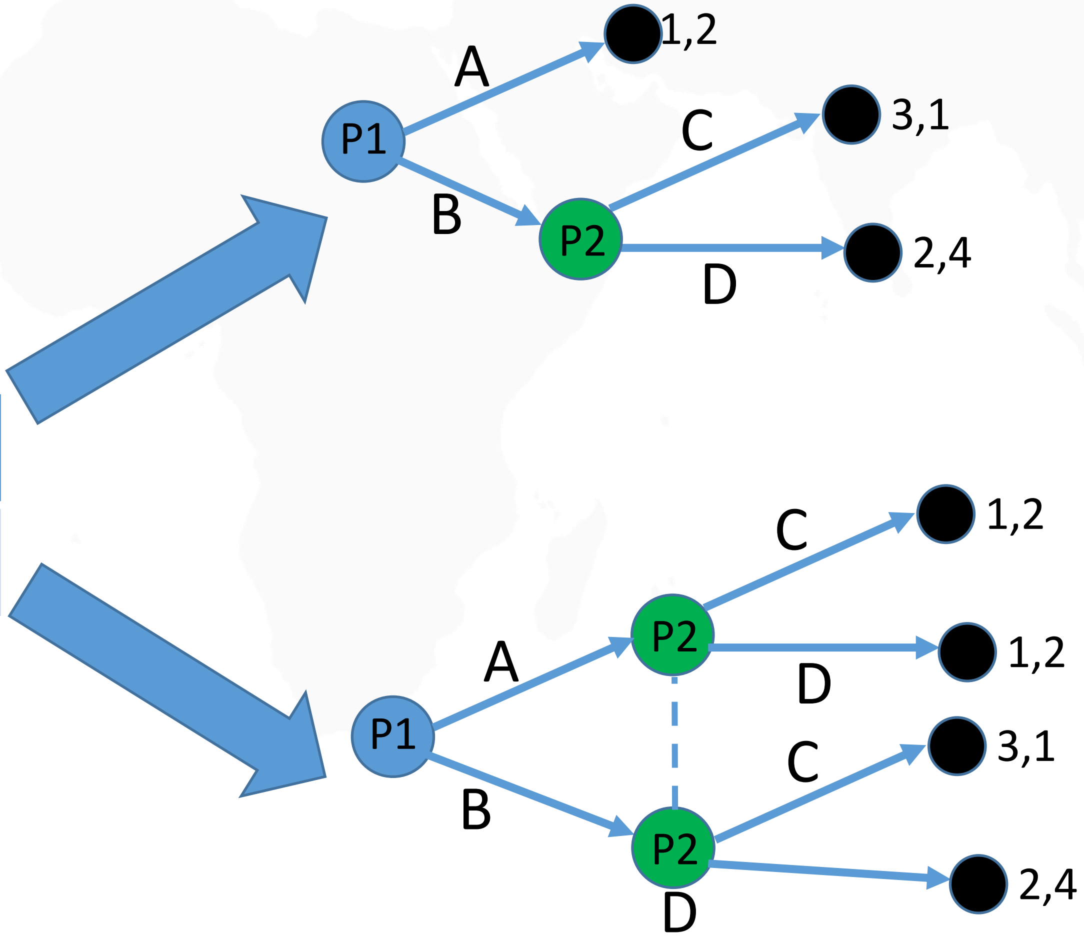
	A	B
UXW	3,3	5,1
UXZ	..	..
UYW	..	..
UYZ	..	..
DXW	..	..
DXZ	..	..
DYW	..	..
DYZ	3,0	1,1

矩阵式表示

矩阵式表示和扩展式表示的转换

- 扩展式转化为矩阵式是唯一的
- 矩阵式转化为扩展式可能不唯一！

	C	D
A	1,2	1,2
B	3,1	2,4



扩展式博弈表示的形式化

博弈展开式表示的形式化描述

展开式博弈可以表示成一个多元组$\Gamma = \langle N, (A_i)_{i\in N}, \mathbb{H}, P, (\mathbb{I}_i)_{i\in N}, (\mu_i)_{i\in N}\rangle$，其中：

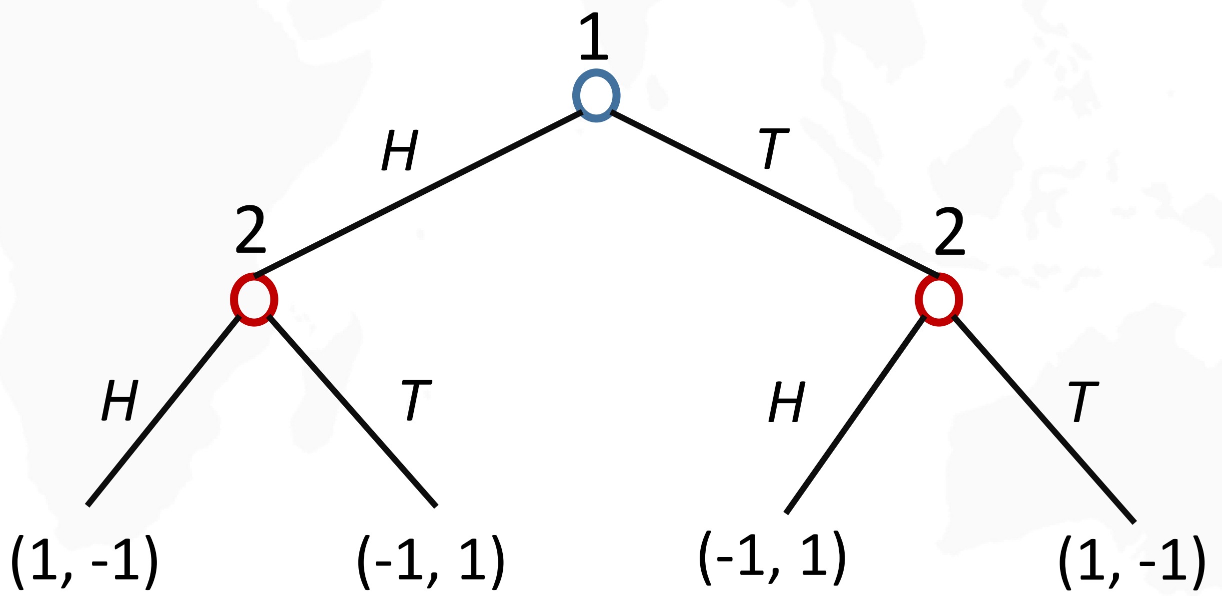
- $N = \{1,2,\cdots,n\}$是参与人集，这是一个有限集合；
- A_i是参与人i的行动集；
- $\mathbb{H}$是由所有终止历史（terminal history）组成的集合，其中一个终止历史是一条从根节点到叶节点的行动路径；以$S_{\mathbb{H}}$表示由所有终止历史的所有真子历史（包括空历史ϵ）组成的集合；
- $P: S_{\mathbb{H}} \to \{1,\cdots,n\}$是一个参与人函数（player function），它将每个真子历史与相应的人联系在一起；
- $\mathbb{I}_i (i = 1,2,\cdots,n)$是由参与人$i$的所有信息集组成的集合；
- $\mu_i: \mathbb{H} \to \mathbb{R}\ (i = 1,2,\cdots,n)$给出了参与人$i$每个终止历史的效用。

扩展式博弈表示的形式化

- 以右下图中的硬币配对博弈为例，说明上述形式化对应的含义：

$\Gamma = \langle N, (A_i)_{i\in N}, \mathbb{H}, P, (\mathbb{I}_i)_{i\in N}, (\mu_i)_{i\in N} \rangle$

  - $N = \{1,2\}$； $A_1 = A_2 = \{H, T\}$；
  - $\mathbb{H} = \{(H,H), (H,T), (T,H), (T,T)\}$， $S_{\mathbb{H}} = \{\epsilon, H, T\}$；
  - $P(\epsilon) = 1$， $P(H) = 2$， $P(T) = 2$；
  - $\mathbb{I}_1 = \{\{\epsilon\}\}$， $\mathbb{I}_2 = \{\{H\}, \{T\}\}$；
  - $\mu_1(HH) = 1, \mu_1(HT) = -1,$
    $\mu_1(TH) = -1, \mu_1(TT) = 1;$
  - $\mu_2(HH) = -1, \mu_2(HT) = 1,$
    $\mu_2(TH) = 1, \mu_2(TT) = -1;$



可观察情形下的硬币配对（参与人1先行动）

第三讲 完全信息动态博弈

内容提纲

1. 博弈类型定义及表示
2. **扩展式博弈均衡分析**
3. 子博弈精炼纳什均衡
4. 扩展式博弈示例讲解

扩展式博弈的纳什均衡

- 分析思路：数学家的思维，尝试先把问题转化为一个已经解决的问题。

待求解扩展式博弈的纳什均衡

分析纳什均衡在扩展式的合理性

先将扩展式转化为策略式表示

在策略式博弈中求解纳什均衡

扩展式博弈的纳什均衡定义

- 通过将扩展式博弈转换为策略式表示，每一个策略组合（也就是博弈树上的一条路径）决定了一个支付向量$u(u_1, \ldots, u_n)$，如果策略组合s^*对于所有的i，能够最大化$u_i(s_i, s_{-i}^*)$（或期望值），即：

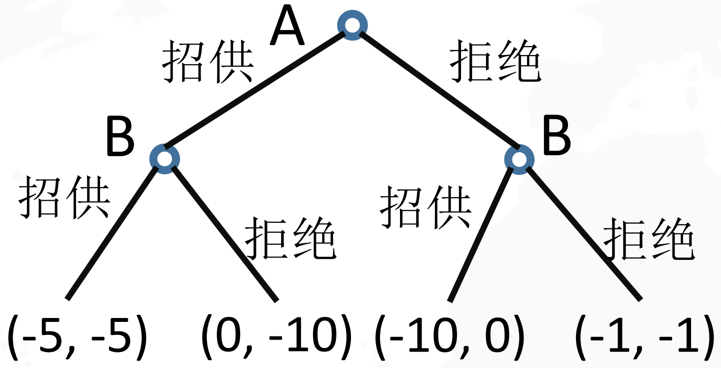
$$s_i^* \in \underset{s_i \in S_i}{\operatorname{argmax}}\, u_i(s_i, s_{-i}^*), \qquad \forall i$$

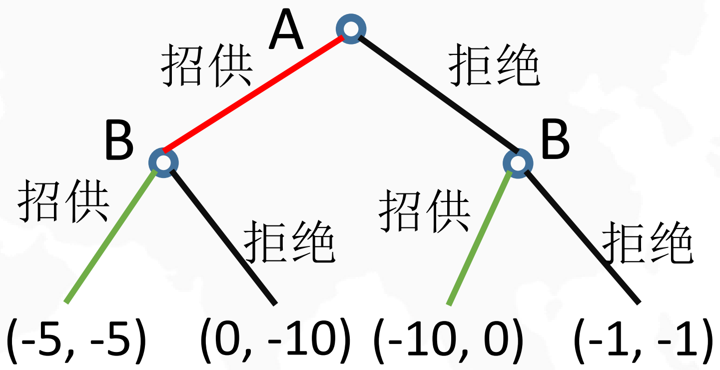
那么该策略组合就是扩展式博弈的纳什均衡。

- 运用类似上述构造方式，对于扩展式表示的博弈的混合策略纳什均衡，可以从期望的角度对其纳什均衡进行构造。

转化法求解扩展式博弈纳什均衡示例

- 用策略转换法求囚徒困境的纳什均衡





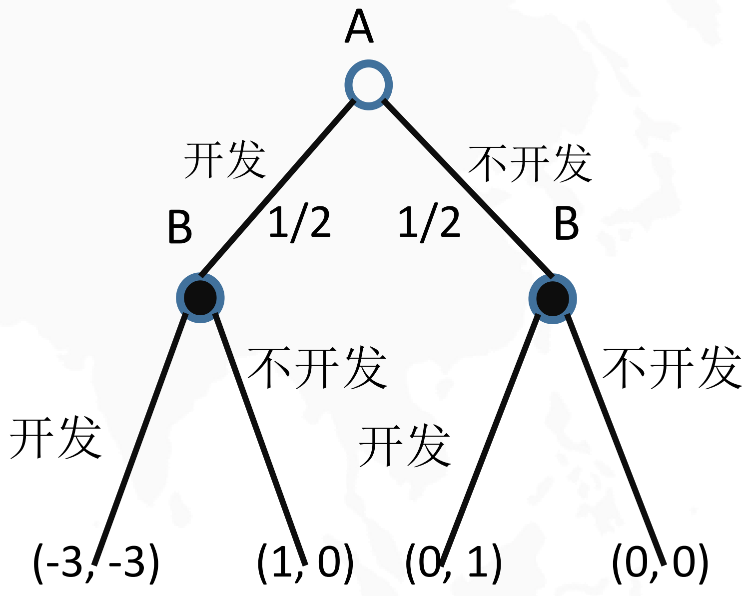
- 扩展式表示的策略
  - A的策略：s_{11} “招供”，s_{12} “拒绝”；
  - B的策略：
    - s_{21}：“无论A是否招供，我都招供”
    - s_{22}：“无论A是否招供，我都拒绝”
    - s_{23}：“A如果招供我就招供，否则拒绝”
    - s_{24}：“A如果招供我就拒绝，否则招供”

策略式表示得到的均衡（s_{11}, s_{21}）

B \ A	s_{21}	s_{22}	s_{23}	s_{24}
s_{11}	**-5**, **-5**	**0**,-10	-5,**-5**	**0**,-10
s_{12}	-10, **0**	-1, -1	**-1**,-1	-10,**0**

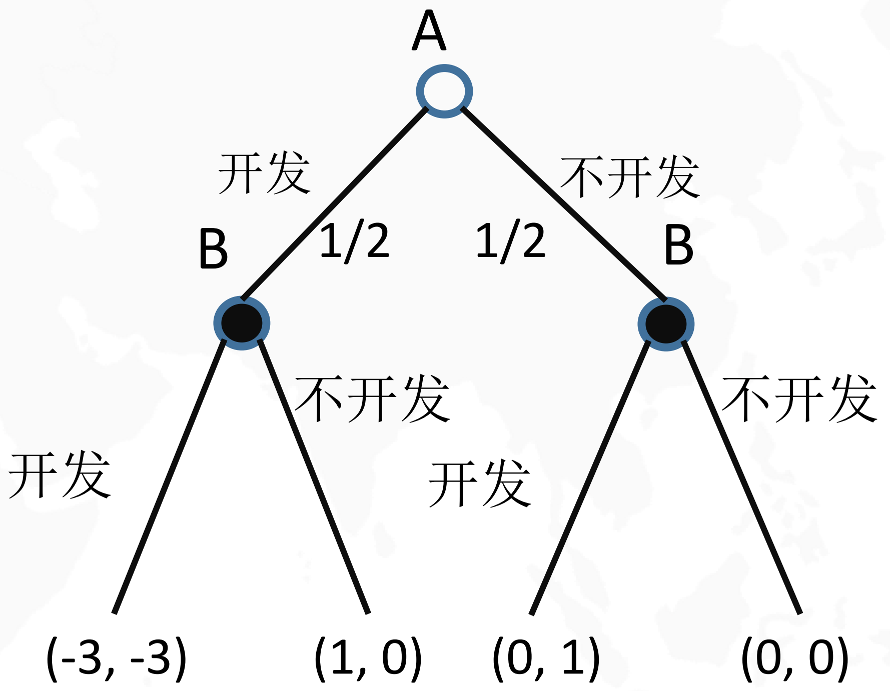
转化法求解扩展式博弈纳什均衡示例

- 卖家市场的房地产开发博弈
  - 低需求，A先做行动，B看到A行动后行动
  - 如右图扩展式表示，完美信息博弈
- 利用信息集的概念进行构造
  - A有一个单节点信息集，两个可选行动
  - B有两个双节点信息集，每个有两个可选行动
- A的纯策略集很简单：（开发、不开发）
- B的纯策略集有四个：{开发，开发}、{开发，不开发}、{不开发，开发}、{不开发，不开发}



转化法求解扩展式博弈纳什均衡示例

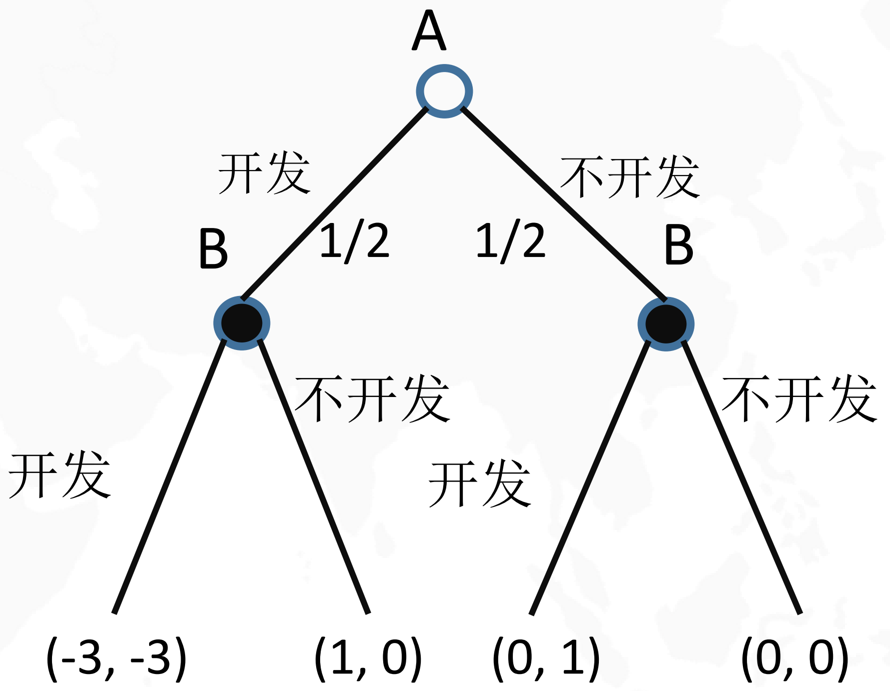
- A的2个纯策略：开发，不开发
- 将B的4个纯策略简记为
  - {开发，开发}：总是要开发
  - {开发，不开发}：跟着干
  - {不开发，开发}：唱反调
  - {不开发，不开发}：总是不开发
- 那么A和B之间博弈的标准式如下



开发商B / 开发商A	{开发，开发}	{开发，不开发}	{不开发，开发}	{不开发，不开发}
开发	-3, -3	-3, -3	**1**, **0**	**1**, **0**
不开发	**0**, **1**	0, 0	0, 1	0, 0

转化法求解扩展式博弈纳什均衡示例

- 右图的房地产开发：
  - 该策略式表示有三个纳什均衡
    - （不开发，{开发，开发}）
    - （开发，{不开发，不开发}）
    - （开发，{不开发，开发}）



只有（开发，{不开发，开发}）是唯一的合理均衡！

开发商B \ 开发商A	{开发，开发}	{开发，不开发}	{不开发，开发}	{不开发，不开发}
开发	-3, -3	-3, -3	**1**, **0**	**1**, **0**
不开发	**0**, **1**	0, 0	0, 1	0, 0

递归法求解扩展式博弈的纳什均衡

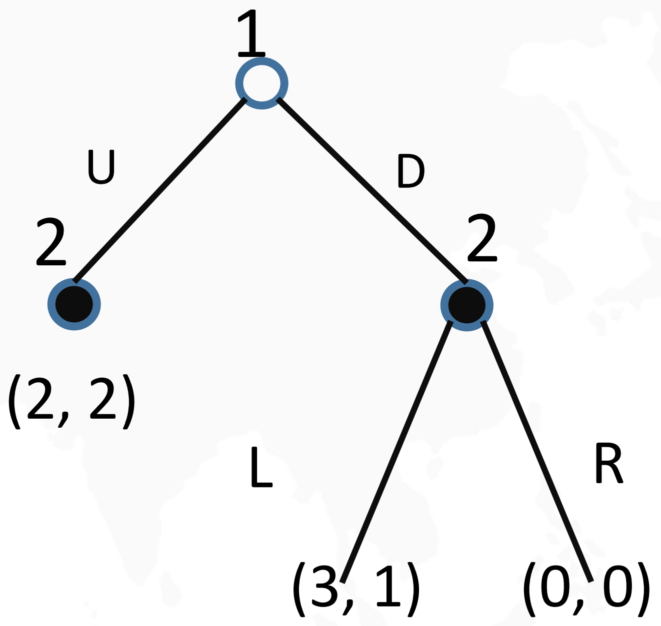
- 扩展式博弈纳什均衡的存在性：有限完美信息博弈

有限完美信息动态博弈纳什均衡（Zermelo,1913; Kuhn, 1953）

每一个有限完美信息动态博弈有一个纯策略纳什均衡。

证明：使用逆向递归法进行求解。
博弈有限：存在一个最后决策结点的集合（倒数第二个结点）
在最后决策结点参与人决策，最大化自己的收益；
给定这个收益，倒数第二个决策结点参与人决策，最大化自己的收益；
如此重复，直到初始结点；
这个倒推过程结束，得到一个**路径**，给出每个参与人的特定策略，所有这些策略构成了一个纳什均衡。

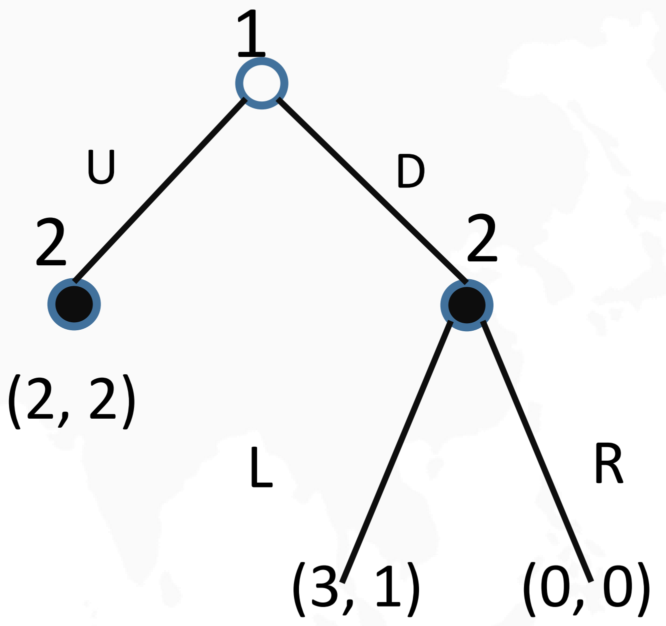
递归法求解扩展式博弈均衡示例



- 扩展式博弈的纳什均衡求取
  - 参与人1先行动，规则如右图如果参与人1选择U，那么无论参与人2选择什么，收益是不变的（2，2）
  - 求取这个博弈的纳什均衡
- 分析过程：
  - 当参与人1选择D时，参与人2有两个选择：如果选择L，参与人2得到1个单位奖励，如果选择R，他能得到0个单位的奖励。
  - 因此L是参与人2的最优选择。
  - 如果参与人2知道参与人1是理性的，那么参与人1将选择D。
  - 因此，(D,L)是一个纯策略纳什均衡

递归法求解扩展式博弈均衡示例

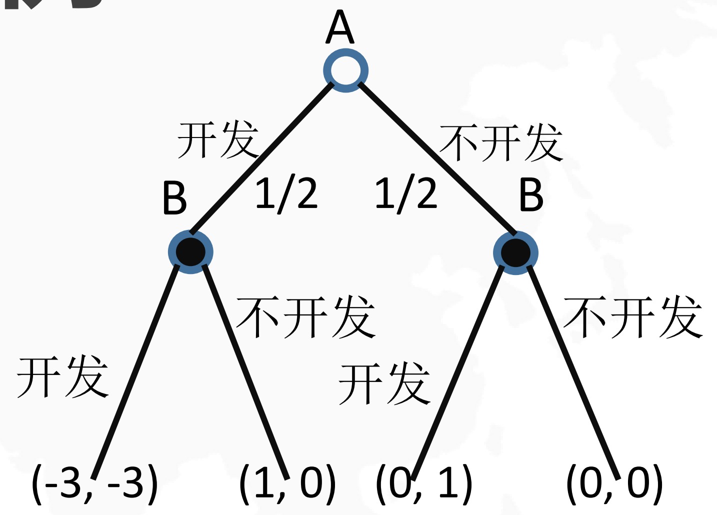
- 扩展式博弈的纳什均衡求取
  - 将上一页的博弈转换为策略式表示：
  - 参与人1 有两个动作：U, D
  - 参与人2 有两个动作：L, R
- 均衡求取：
  - 重复剔出劣策略
  - 对参与人2，来说R是一个弱劣策略。
    - 即，无论参与人1选择U还是D, R 都不是参与人2的优策略。
  - 重复剔除之后即可得到和前面分析相同的结果



参与人1 \ 参与人2	L	R
U	2,2	2,2
D	**3,1**	0, 0

- 结论：逆向归纳法实际是重复剔除劣策略在扩展式博弈中的应用。

递归法求解扩展式博弈均衡示例



- 卖家市场的房地产开发博弈
  - 低需求，A先做行动，B看到A行动后行动
  - 如右图扩展式表示，完美信息博弈
- 使用递归法求解：从最后一个子博弈开始分析，直到分析完整个博弈，记录其中所有子博弈均能保持纳什均衡的那些路径，都是整个扩展式博弈的纳什均衡。
  - A选择开发那条路径：B的均衡决策是不开发
  - A选择不开发那条路径：B的均衡决策是开发
  - 因此，作为理性人的A，肯定知道B是理性的，所以A在一开始的均衡策略是开发，整个博弈的均衡策略是{开发，不开发}

递归法求解扩展式博弈均衡示例

- 通过逆向递归法可求出有限完美信息的扩展式博弈的纳什均衡解。
- 对于无限的扩展式博弈或者不完美信息扩展式博弈，逆向归纳法并不适用于所有情况：
  - 不完美信息博弈
    - 信息集不是单结的，逆向递归法没法直接对非单结信息集进行分析
  - 无限博弈：不知道最后一个子博弈在哪里。
    - 一个决策结有无穷多个后续结，如果支付函数不是特殊限制，那么最优选择不一定存在；或一个路径包含无穷多个决策结，不存在最后一个决策结。
  - 逆向归纳法不适用，并不代表无限不完美信息博弈均衡不存在。
- 逆向递归法给出的完全信息动态博弈的均衡，被称为**子博弈精炼纳什均衡**，下一节详细讲解。

第三讲 完全信息动态博弈

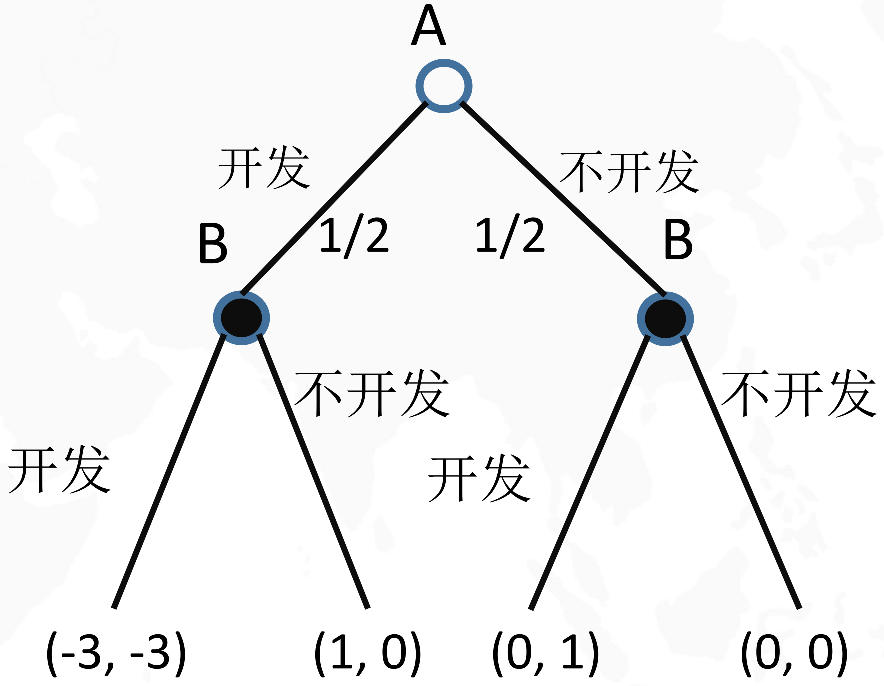
内容提纲

博弈纳什均衡的“精炼”

- 动态博弈的扩展式表示都可以转换为策略式表示，从而转换为静态博弈。
- 完全信息静态博弈的纳什均衡因此同样适用于动态博弈。
- 但是这样得到的（多个）纳什均衡往往不是合理的均衡。
  - 纳什均衡会有多个均衡
  - 未考虑到参与者行动的先后顺序，使得很多均衡不合理
- 需要寻求能够改进（perfecting）和精炼（refining）纳什均衡概念，从而得到更为合理的博弈解。
- 精炼的含义：把动态博弈中的“合理纳什均衡”和“不合理纳什均衡”分开

子博弈精炼纳什均衡

- 以右图的房地产开发为例：
  - 首先转化为下图的策略式表示
  - 该策略式表示有三个纳什均衡
    - (不开发，{开发，开发})
    - (开发，{不开发，不开发})
    - (开发，{不开发，开发})



只有（开发，{不开发，开发}）是唯一的合理均衡！

开发商B / 开发商A	{开发，开发}	{开发，不开发}	{不开发，开发}	{不开发，不开发}
开发	-3, -3	-3, -3	**1, 0**	**1, 0**
不开发	**0, 1**	0, 0	0, 1	0, 0

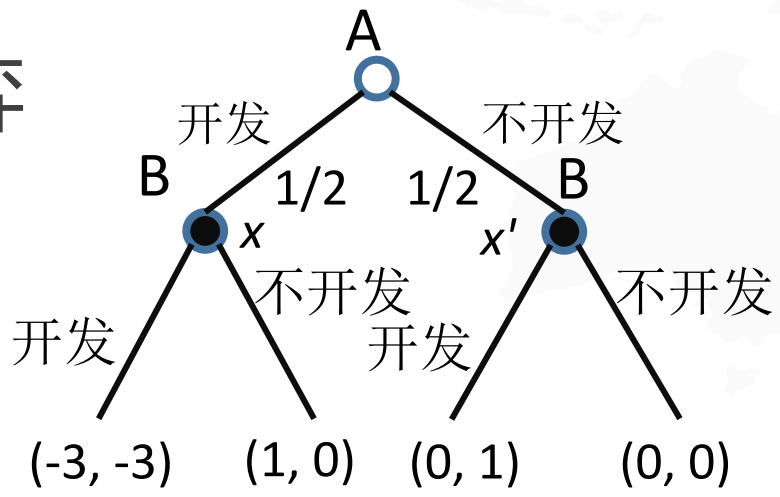
子博弈精炼纳什均衡

- 子博弈：原博弈的一部分，本身可以作为一个独立博弈进行分析。

子博弈

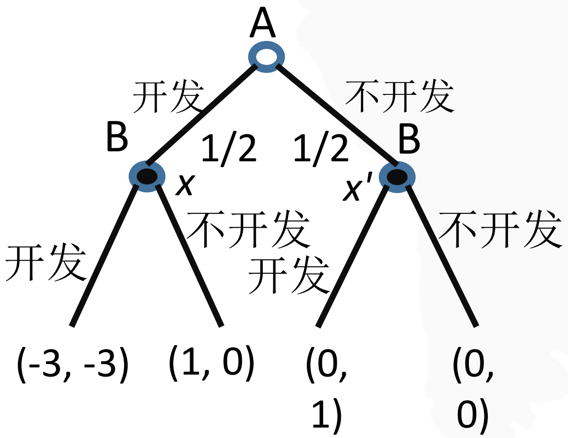
一个扩展式表述博弈的子博弈G由一个决策结x和所有该决策结的后续结$T(x)$（包括终结点）组成，它满足下列条件：1）x是一个单节点信息集，即$h(x)=\{x\}$；2）对于所有的$x' \in T(x)$，如果$x'' \in h(x')$，那么$x'' \in T(x)$。

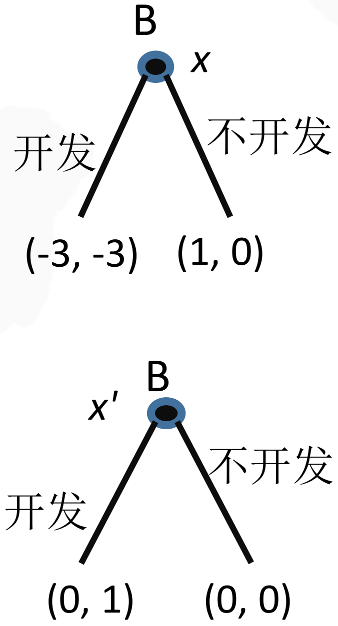
- 任何博弈本身都可以称为自身的一个子博弈

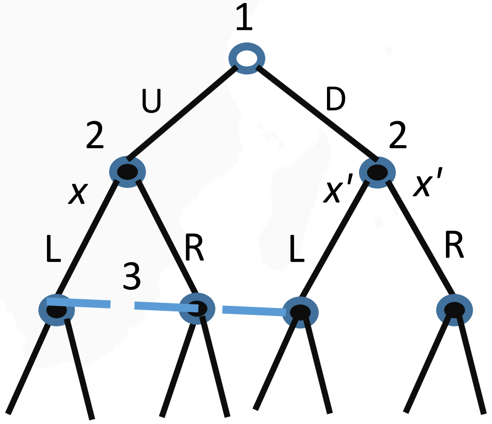


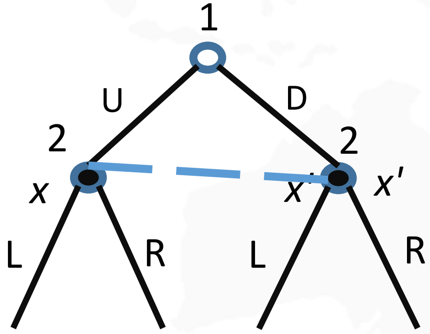
子博弈精炼纳什均衡

- 子博弈的两个条件分析
  - **条件一**：一个子博弈必须从一个单结信息集开始
  - **条件二**：子博弈根节点的每个后续节点所在信息集的所有节点都必须在子博弈中
  - 另外：子博弈信息集和支付向量需要都直接继承自原博弈









子博弈精炼纳什均衡

- 子博弈精炼纳什均衡：一个策略组合是子博弈精炼纳什均衡，当且仅当它在每一个子博弈（包括原博弈）上都构成一个纳什均衡。

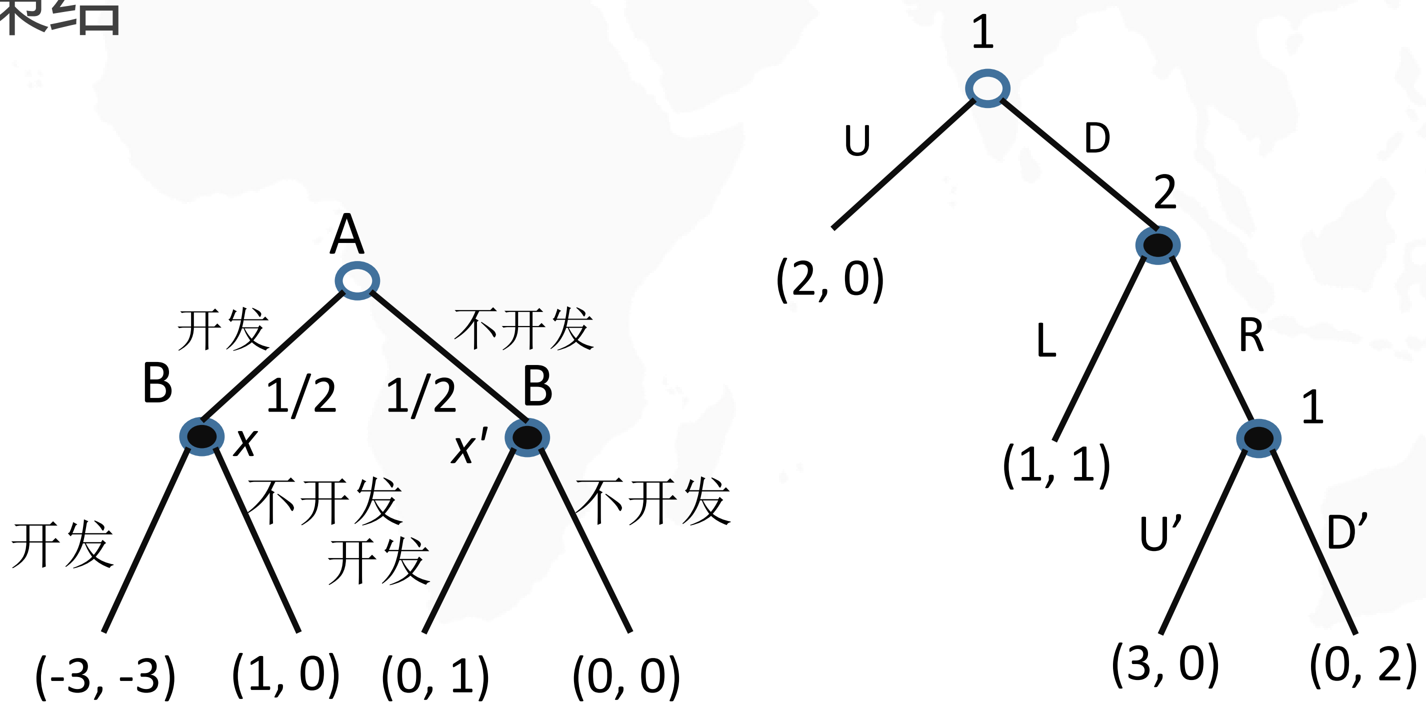
子博弈精炼纳什均衡（Selten,1965）

扩展式描述博弈的策略组合$s^* = (s_1^*, \ldots, s_i^*, \ldots, s_n^*)$是一个子博弈精炼纳什均衡，如果：1）它是原博弈的纳什均衡；2）它在每一个子博弈上给出纳什均衡。

- 在每一个子博弈上给出纳什均衡：无论过去发生什么事情，参与人在每一个决策结上都应该最优化自己的决策。

子博弈精炼纳什均衡

- 使用逆向归纳法求解子博弈精炼纳什均衡
  - 从博弈的最后一个决策结出发，该决策结上行动的参与人做最优决策
  - 倒回到第二个决策结，找出倒数第二个决策者的最优决策
  - 不断重复直到初始决策结
- 举例：
  - 房地产开发博弈
  - 三阶段行动博弈



第三讲 完全信息动态博弈

内容提纲

1 博弈类型定义及表示

2 扩展式博弈均衡分析

3 子博弈精炼纳什均衡

4 扩展式博弈示例讲解

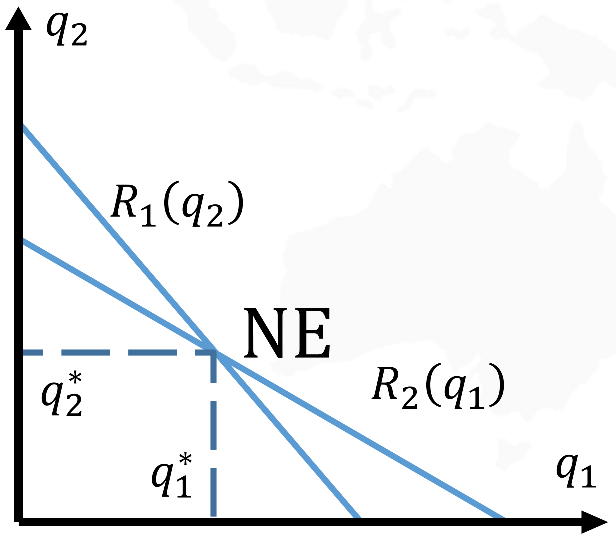
应用示例1：Stackelberge寡头竞争模型

- 斯坦克尔伯格（Stackelberge）寡头竞争模型
  - 企业的行动有先后顺序，领头企业1首先选择产量$q_1 > 0$，尾随企业2根据领头企业的产量选择自己的产量$q_2 > 0$。
  - 博弈类型：完美信息**动态**博弈
  - 博弈树：领头企业作为初始决策结，尾随企业有无穷多个决策结
- 对比库诺特（Cournot）寡头竞争模型
  - 两个企业同时决策；博弈类型属于完全信息**静态**博弈

$$\begin{cases} q_1^* = R_1(q_2) = \frac{C_1'(q_1)}{P'(q_1+q_2)} - \frac{1}{P'(q_1+q_2)} P(q_1+q_2) \\ q_2^* = R_2(q_1) = \frac{C_2'(q_1)}{P'(q_1+q_2)} - \frac{1}{P'(q_1+q_2)} P(q_1+q_2) \end{cases}$$ 反应函数

赋值：需求函数：$P = a - q_1 - q_2$ 单位成本：$c > 0$

结果是：$q_1^* = q_2^* = (a-c)/3$



应用示例1：Stackelberge寡头竞争模型

- 斯坦克尔伯格（Stackelberge）寡头竞争模型
  - 企业的行动有先后顺序，领头企业1首先选择产量$q_1 \geq 0$，尾随企业2根据领头企业的产量选择自己的产量$q_2 \geq 0$。
  - 价格函数：$P = a - q_1 - q_2$ 单位成本：c>0
- 求解：
  - 首先给定q_1求企业2的最优选择：$\max\limits_{q_2 \geq 0} \pi_2(q_1, q_2) = q_2(a - q_1 - q_2) - q_2 c$
  - 最优化一阶条件（对q_2求导）得：$s_2(q_1) = (a - q_1 - c)/2$
  - 企业1知道企业2是理性的（知道其尾随的最优决策是$s_2(q_1)$）：$\max\limits_{q_1 \geq 0} \pi_1(q_1, q_2) = q_1\big(a - q_1 - s_2(q_1)\big) - q_1 c$
  - 同样求一阶条件得：$q_1^* = (\mathrm{a} - \mathrm{c})/2$；进而得：$q_2^* = s_2(q_1^*) = (\mathrm{a} - \mathrm{c})/4$

应用示例1：Stackelberge寡头竞争模型

- 对比斯坦克尔伯格均衡结果与库诺特均衡结果

	库诺特	斯坦克尔伯格
博弈类型	完全信息静态博弈	完全信息动态博弈
行动先后	同时行动	领头企业先动
纳什均衡	$(q_1^* = r(q_2), q_2^* = r(q_1))$	$(q_1^*, s_2(q_1))$
均衡结果	$((a-c)/3, (a-c)/3)$	$((a-c)/2, (a-c)/4)$
总产量	$2(a-c)/3$	$3(a-c)/4$
价格（P）	$(a+2c)/3$	$(a+3c)/4$
利润	$((a-c)^2/9, (a-c)^2/9)$	$((\boldsymbol{a}-\boldsymbol{c})^{\mathbf{2}}/\mathbf{8}, (a-c)^2/16)$

应用示例1：Stackelberge寡头竞争模型

- 结果分析：
  - 相较于库诺特均衡，企业1在斯坦克尔伯格均衡中的利润增加了，而企业2的利润减少了，我们称之为**先手优势**。
  - 这种先手优势的背后，企业2拥有更多的信息却使其在博弈中处于劣势。
  - 如果在博弈之前，企业1和企业2都不知道彼此的生产信息，那么企业1的产量威胁就是一种口头宣扬，不可置信。
  - 一旦在博弈之前，企业1知道了企业2会根据他的产量来做决策，那么他会把“口头”威胁，变成实际行动，这种威胁就是可置信的，迫使博弈走入斯坦克尔伯格均衡。

应用示例2：RS议价模型

- 鲁宾斯坦恩（Rubinstein）— 斯塔尔（Stahl）议价模型
  - 参与人1和参与人2共同分享一块蛋糕
  - 第一个时期，参与人1先给出一个分享规则$(x_1, 1-x_1)$
  - 参与人2选择接受，则结束；选择拒绝，则进入第二个时期
  - 第二个时期，参与人2给出一个分享规则$(x_2, 1-x_2)$
  - 参与人1选择接受，则结束；若拒绝，则进入第三个时期
  - ……
  - 直到有人接受对方的规则，则停止
- 分析：博弈类型为无限期完美博弈，但有可能被中止；收益函数为$\pi_i = \delta_i^{t-1} x_t$，其中$\delta_i$是参与人$i$的贴现因子

应用示例2：RS议价模型

- 鲁宾斯坦恩（Rubinstein）— 斯塔尔（Stahl）议价模型
  - 分析思路：先从有限次T内可以结束博弈开始($T < \infty$)，然后讨论与贴现因子δ和博弈次数T之间的关系。
- 用逆向递归法求解有限次情况下的子博弈精炼纳什均衡
  - $T = 2$，(只进行两次)：
    - 如果在t = 2时参与人2提出的规则是$(x_2, 1 - x_2)$, 其中$x_2 = 0$, 即$(0,1)$, 参与人1也会接受，因为参与人1没有机会进行下一次提议了。
    - 而参与人2在$t = 2$时得到的1个单位的收益，贴现到$t = 1$时刻为δ_2。
    - 如果在$t = 1$时刻，参与人1提出的规则$(x_1, 1 - x_1)$中，只要$1 - x_1 \geq \delta_2$，参与人2就会接受。
    - 子博弈精炼均衡为$(1 - \delta_2, \delta_2)$.
  - $T = 3$, 最后做决策的是参与人1：同样分析，得到子博弈精炼均衡为$(1 - \delta_2(1 - \delta_1), \delta_2(1 - \delta_1))$
  - ……
  - 接下来讨论δ和T的各种情况

应用示例2：RS议价模型

- 鲁宾斯坦（Rubinstein）— 斯塔尔（Stahl）议价模型
  - 上一页均衡为$\left(1-\delta_2(1-\delta_1(1-\delta_2)),\quad \delta_2(1-\delta_1(1-\delta_2))\right)$
  - 如果$\delta_1=\delta_2=0$：两个人都没有耐心，均衡结果肯定是$(1,0)$
  - 如果$\delta_2=0$：无论δ_1为多少，结果都是$(1,0)$。参与人1知道参与人2不会有耐心进行讨价还价，所以参与人给出最有利于自己的报价。
  - 如果$\delta_1=0,\delta_2>0$：均衡结果是$(1-\delta_2,\delta_2)$，因为参与人2将在第二轮得到最大收益，折现到第一轮中就是δ_2，那么参与人1将报价$(1-\delta_2,\delta_2)$。
  - 如果$\delta_1=\delta_2=1$，无限耐心：$T=1,3,5,\ldots$，均衡为$(1,0)$；$T=2,4,6,\ldots$，均衡为$(0,1)$；后动优势，谁最后一轮出价，谁得到的最多。

应用示例2：RS议价模型

- 鲁宾斯坦恩（Rubinstein）—斯塔尔（Stahl）议价模型
  - 上一页均衡为$\left(1-\delta_2(1-\delta_1(1-\delta_2)),\quad \delta_2(1-\delta_1(1-\delta_2))\right)$
- 如果$0<\delta_i<1$:
  - 均衡结果不仅依赖于贴现因子之间的比率，而且依赖于博弈的时间T和谁在最后一个阶段出价。
  - 这种依赖关系会随着时间的增大而缩小。
  - 当T趋于无穷的时候，我们将得到“先动优势”。
  - 当$\delta_1=\delta_2=\delta$时，我们得到唯一的均衡结果$\left(\frac{1}{1+\delta},1-\frac{1}{1+\delta}\right)$

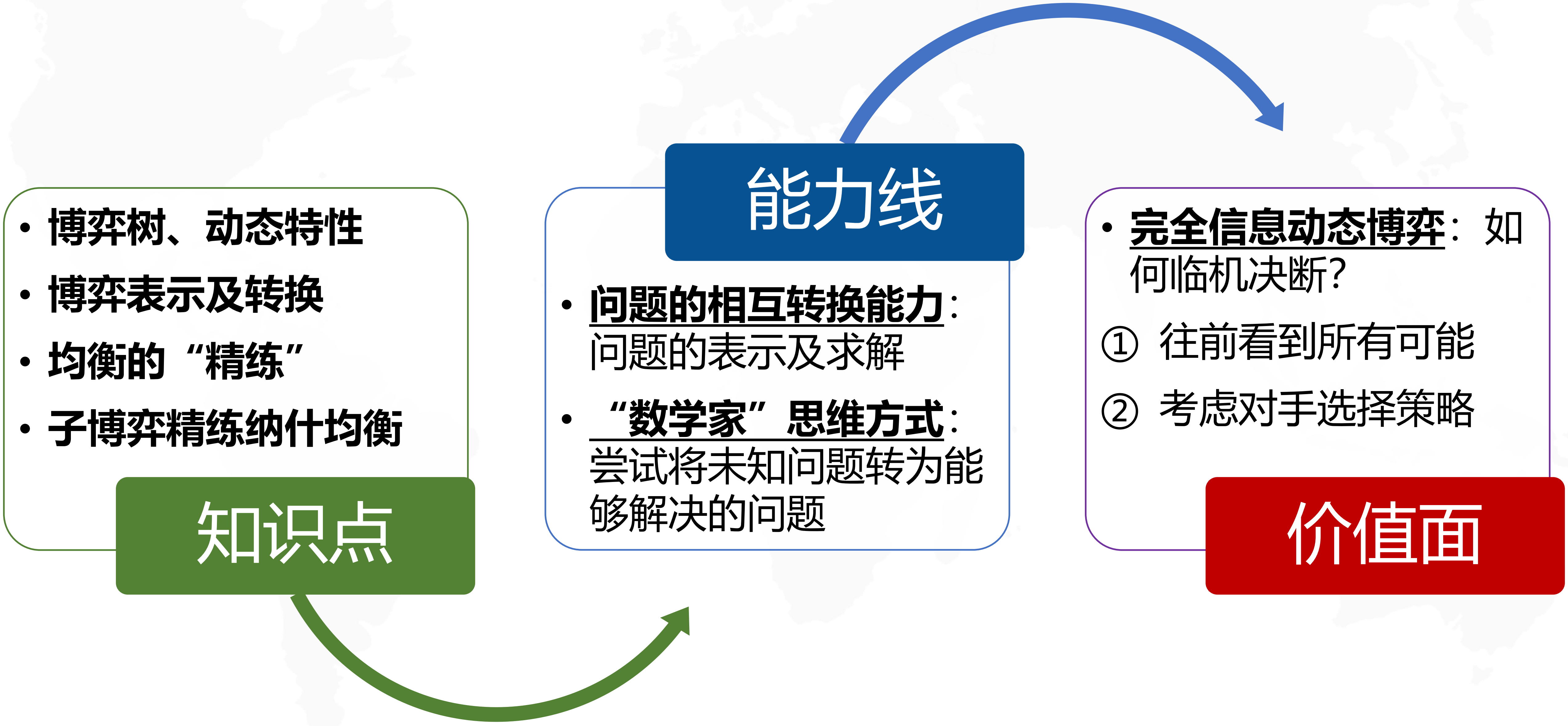
定理(Rubinstein,1982)

在无限期轮流出价博弈中，唯一的子博弈精炼纳什均衡结果是：$x^*=\frac{1-\delta_2}{1-\delta_1\delta_2}$

应用示例2：RS议价模型

- 鲁宾斯坦恩（Rubinstein）— 斯塔尔（Stahl）议价模型
- 定理的简单说明:
  - 当我们讨论无穷轮博弈的时候，可以从任意一个由参与人1起点的子博弈开始讨论。
  - 时刻t，参与人1能得到的最大收益是M；t时刻的M，等价于$t-1$时刻的$\delta_1 M$; 参与人2知道，在$t-1$时刻的任意$x \geq \delta_1 M$的报价都会被接受。
  - 所以$t-1$时刻，参与人2给出方案$(\delta_1 M, 1-\delta_1 M)$；$t-1$时刻得到的$1-\delta_1 M$等价于在$t-2$得到的$\delta_2(1-\delta_1 M)$
  - 所以$t-2$时刻，参与人1报价$\left(1-\delta_2(1-\delta_1 M), \delta_2(1-\delta_1 M)\right)$
  - 无限博弈中，t时刻看到的与$t-2$时刻看到的应该相同：$M=1-\delta_2(1-\delta_1 M)$，解得 $M=\frac{1-\delta_2}{1-\delta_1\delta_2}$

本讲内容小结



博弈树、动态特性
博弈表示及转换
均衡的“精练”
子博弈精练纳什均衡
知识点
能力线
问题的相互转换能力：问题的表示及求解
“数学家”思维方式：尝试将未知问题转为能够解决的问题
完全信息动态博弈：如何临机决断？
① 往前看到所有可能
② 考虑对手选择策略
价值面

本次课程作业

- 作业内容：寻找存在子博弈精炼纳什均衡的完全信息动态博弈习题两道，给出习题描述、问题分析、求解过程和最终结果。
  - 评分准则：选取习题的典型性、分析求解过程的完整性以及最终结果的指导性意义。
- 截止时间：下次上课前（2024年3月18日）
- 提交方法：在Sep课程网站上提交，同时提交电子版到助教邮箱（xuhang2020@ia.ac.cn）
- 邮件发送规范
  - 邮件主题：博弈论第**三**次作业_**学号**_**姓名**
  - 附件名称：博弈论第**三**次作业_**学号**_**姓名**.docx

感谢大家认真听讲

兴军亮

jlxing@nlpr.ia.ac.cn

2024年3月11日