博弈论第九讲

授课时间：2021 年 11 月 12 日　授课教师：张家琳　记录人：王柏森　王卓

1 投票机制 (Voting Mechanism)

对于投票机制，先回顾上节课的社会福利机制。

1.1 社会福利机制 (Social welfare)

定义 1 (社会福利函数 (Social welfare function)). *定义 $F: L^n \to L$，我们希望社会福利函数满足：*

- *一致性*
- *非独裁*
- *独立性*

定理 2 (阿罗不可能定理). *任意参选人超过 2 人的社会福利函数，如果满足一致性和独立性，则一定是独裁的。*

上一节课并没有证明完全，我们先简单回顾一下之前的证明思路。

1. 证明一个引理：如果所有人的序关系是极端的，得到的结果也是极端的。
2. 固定极端的 X，构造一类特殊的投票 $\mathcal{Q}^{(i)}$ 。
3. 在投票中找到潜在的独裁者 i 。并证明最终的结果中，不包含 X 的二元关系在 i 的投票和最终投票结果中保持一致。
4. 证明包含 X 的二元关系在 i 的投票和最终投票结果中也保持一致，从而证明我们构造的机制是独裁的。

1-3 的证明可根据 lecture note 8，本节课开始对 4 的证明。

在证明之前，我们先给出一些符号的定义。注意：**一些定义与 note 8 并不一致！**因为 note8 中一些符号使用较为混乱。

- S：参选人集合
- L：A 上的全序关系
- $F: L^n \to L$：社会福利函数
- $\prec_L$：L 对应的二元序关系
- $\mathcal{Q}$：所有投票者的一次投票
- $\mathcal{Q}_i$：投票 Q 中 i 投出的全序关系

- $F(\mathcal{Q})$：F 在投票 $\mathcal{Q}$ 中的选举结果
- 其余的大写字母指参选者（A，X 等），小写字母为投票者（i，j 等）
- $i < j$：Step2 中的投票者的次序关系中，i 的次序早于 j
- $\mathcal{Q}^{(i)}$：Step2 中构造的一族特殊投票

我们的初步证明思路是首先固定 X，找到对 X 的潜在独裁者 i；再对于任意的 $Y \neq X$，找到潜在独裁者 j。最终目标证明 $i = j$ 即可。即：对于任意投票 $\mathcal{Q}$ 和 $A \neq X$，如果 $A \prec_{\mathcal{Q}_i} X$，则 $A \prec_{F(\mathcal{Q}_i)} X$。

证明

1. 当参选者数量大于三，即 $|S| > 3$ 时，不妨记 $S = \{X, Y, A, B \dots\}$。X 对应的潜在独裁者为 i，Y 对应的潜在独裁者为 j。根据 Step3，i 可以满足任意非 X 的 A，B 的序关系，j 也可以满足任意非 Y 的 A，B 的序关系，则 $i = j$。
2. 只有三个人时，不妨设 $S = \{X, Y, Z\}$，X 对应的潜在独裁者为 i, Y 对应的潜在独裁者为 j。假设 $i \neq j$，由对称性，不妨令 $j < i$，由定义及第三步证明可知：对于任意的 $k < i$，X 均在 $F(\mathcal{Q}^{(k)})$ 中排名第一，因此有：$X \prec_{F(Q^{(j)})} Y$，但根据潜在独裁者定义，$Y \prec_{Q_j^{(j)}} X$，与 j 是 Y 对应的潜在独裁者矛盾！原结论成立。

综上，原结论成立。 □

据此与 Step3，我们得到了 $F(Q)$ 与 Q_i 中任意序关系都相同，我们就证明了 F 是独裁的，从而证明了阿罗不可能定理。

1.2 社会选择机制 (Social choice)

社会选择机制，则是通过投票从 $|A|$ 名候选人中选出一个当选者。

定义 3 (社会选择函数 (Social choice function))**.** *定义 $F : L^n \to A$，我们希望社会选择函数满足：*

- *一致性 (unanimity)*
- *激励相容 (incentive compatible)*
- *非独裁 (not dictatorship)*

定义 4 (激励相容 (Incentive compatible))**.** *一个社会选择函数 F 能够被投票者 i 策略地操纵 (strategically manipulated) 是指存在某个投票者的全序列 $\prec_1, \prec_2, \cdots, \prec_n$，以及 $\prec_i'$，使得 $F(\prec_1, \cdots, \prec_i, \cdots, \prec_n) \prec_i F(\prec_1, \cdots, \prec_i', \cdots, \prec_n)$，即 i 可以通过谎报自己的全序使得自己更喜欢的候选人当选。而 F 是激励相容的就是说它不能被任何投票者 i 策略操纵。*

定理 5 (吉伯德-萨特思韦特不可能定理)**.** *对于一个满足激励相容约束和一致性的社会选择机制，若候选人集合为 A，当 $|A| \geq 3$ 时，该机制中一定存在一位独裁者。*

此定理的证明基于上节课的阿罗不可能定理的证明。

2 无金钱的机制设计 (Mechanism Design without Money)

在许多重要的机制设计中，例如投票、器官捐赠、择校等，金钱的参与是不可行的或不能接受的，此时设计不涉及金钱的机制则十分重要。

2.1 房屋分配 (House Allocation)

例 1 [房屋分配] 共有 n 名参与者，n 间房屋，对于每一位参与者，都有一个私人的对 n 间房屋的偏好，设计一个尽可能满足所有人偏好的分配机制。

解 [首位交易环算法 (Top Trading Cycles Algorithm, TTCA)]

先任意地为所有参与者分配一个房屋，构造有向图 G，所有参与者与其房子的二元组构成顶点集 S：

1. 定义 $T = S$。
2. 边集定义：对于每一条有向边 $e = (i, j)$，意为 j 当前最喜欢 i 的房子。则 e 构成边集 E。
3. 找到一条有向环。（此时节点数 $|T|$，边数 $|T|$，由离散数学知识可知图中一定有有向环）
4. 将环里的房子进行分配。设环为 $N = \{c_1, \ldots, c_n\}$，环中 $c_1 \to c_2 \to \cdots \to c_n \to c_1$。则 c_i 将被分配到 c_{i+1} 的房子，且 c_n 将被分配到 c_1 的房子。
5. 将分出去的房子所对应的二元组从 T 中删去。
6. 重复，跳转至 2，直到 T 为空。

2.1.1 TTCA 算法性质

- 可终止性 (Termination)

  TTCA 算法可以终止。

  证明 TTCA 第二步形成的有向图中每个节点的出度都是 1，所以一定会形成环，因此 TTCA 每次循环都会有参与者重新分配到房间，至多循环 n 次，每名参与者都会重新分配到房间，所以该算法一定可以终止。 □

- 弱改进分配 (Weakly improved allocation)

  TTCA 算法得到的结果一定不比原来差。

  证明 对任意一名参与者，因为自己与自己的房子形成自圈，所以要么他在形成自圈前获得房子，要么在自圈出现时获得房子。根据定义，在新分配方案中，每一位参与者得到的房间一定不会比开始差。 □

- 激励相容性 (Incentive compatibility)

  TTCA 算法中，当每个人对房子的偏好是私有信息时所有人没有说谎的动机。这是房屋分配机制必须要存在的关键特性。

证明 对于任一参与者，假设他在第 k 轮重新获得房间。首先，在第 k 轮前，他没有与任何其他参与者形成环，而该参与者仅能改变自己的指向，所以无法改变已经形成的环，因此他只能与当前未形成环的其他参与者成环进而改变分配。其次，对于该参与者而言，若想通过欺骗在第 k 轮前与原本不在前 k 轮成环的参与者形成新环完成分配，则根据定义，第 k 轮时当前参与者获得了没有在前 k 轮被分配到的房子中最喜欢的一个，因此他没有动力通过欺骗提前成环，否则结果不会变得更好。综上，TTCA 中的每一位参与者都会诚实选择，所以满足激励相容性。 □

- 硬核分配 (Core allocation)
  TTCA 算法得到的结果中，不存在集合 $C \subset S$，使得集合 C 中的元素通过彼此交换房间，可以使得每个人的房间都变好。

  证明 反证：设 N_i 为 TTCA 算法中得到的第 i 个环。假设 TTCA 得到的不是硬核分配，则 $\exists C \subset S$，则可以找到一个最小的 i，使得 $C \cap N_i \neq \emptyset$，选择 $j \in C \cap N_i$，因为 $C \cap (N_1 \cup \cdots \cup N_{i-1}) \neq \emptyset$，所以 C 中没有 $(N_1, \cdots, N_{i-1})$ 中的房子。而由 TTCA 得到的分配中，j 已经得到了除 $(N_1, \cdots, N_{i-1})$ 之外最好的房子，所以 C 中没有房间可以让 j 变好，由定义矛盾出现，原命题得证。 □

2.2 稳定婚姻 (Stable matching)

定义 6 (不稳定配对 (Unstable pair)). 在一个匹配结果中，对于其中一对 (A, x)，若与当前匹配的同伴相比，A 更喜欢 x，x 也更喜欢 A，则称 (A, x) 为一个不稳定配对。

例 2 [稳定婚姻 (Stable matching)] 有 n 个男孩，n 个女孩，每个人对相对性别的 n 名成员都有一个总的偏好顺序，设计一个稳定匹配机制，使得匹配结果中不存在不稳定配对。

稳定婚姻问题由 Gale 与 Shapley 提出，但解决该问题的算法——延迟接受算法（Deferred Acceptance Algorithm，DAA），却早于该问题的提出时间。但由于稳定婚姻问题，DAA 算法也被称为 Gale-Shapley 算法。我们先给出 Gale-Shapley 算法：

1. 对每个男生进行循环
2. 若存在某个男生 i 当前没有配对，则找到 i 最喜欢的、没有拒绝过 i 的女生 j。否则，终止算法。
3. 若女生 j 当前没有配对，则 (i, j) 配对成功，转到下一个男生。否则转 4。
4. 若女生 j 当前已经配对，设 j 当前与 v 配对，则权力反转，计算女生 j 更喜欢 i 还是 v。若 i 更喜欢 j，则 (i, j) 配对成功，j 拒绝 v；否则 (v, j) 配对成功，j 拒绝 i。转到步骤 2。

简单来说，在 Gale-Shapley 算法中，每个人按照喜欢的顺序依次询问异性是否配对，而被追求者每次都留下当前最喜欢的追求者。

我们来讨论 Gale-Shapley 算法的几个性质。

2.2.1 Gale-Shapley 算法性质

- 可终止性 (Terminate)
  Gale-Shapley 算法可以终止，且每人都有配对。

证明 每位追求者向每位被追求者最多询问一次，每位被追求者最多选择 n 次，因此最多发生 n^2 次询问，因此算法可以终止。

若最终有人没有配对，由于匹配的男女生数量一致，则没有配对的男女数量也一致，则至少有一个男生未匹配。但若男生没有被匹配，仍需执行算法步骤 2，因此算法此时并未终止。 □

- *返回稳定匹配* (Return a stable matching)

  Gale-Shapley 产生的匹配是稳定的。

  证明 反证法：假设从 Gale-Shapley 算法得到的匹配中存在追求者 i 和配偶 j 以及追求者 i' 和配偶 j'，满足 i 在 j 和 j' 中更喜欢 j'，且 j' 在 i 和 i' 中更喜欢 i ，那么 j' 必然拒绝过 i(否则 (i, j') 应该为一对)，而 Gale-Shapley 算法中被追求者配偶的质量是单调上升的（因为每次都选择更喜欢的一位），因此对 j' 而言，i 和 i' 中更喜欢 i'，矛盾！原结论得证。 □