

Multi-Parameter Mechanism Design

Joseph Chuang-Chieh Lin

Dept. CSIE, Tamkang University, Taiwan



- ▶ In previous lectures, we only consider single-parameter mechanism design problems.
 - ▶ The only private parameter of an agent is her valuation.
- ▶ In this lecture, we:
 - ▶ Introduce multi-parameter environments, where each agent has multiple private parameters.
 - ▶ Introduce the Vickrey-Clarke-Groves (VCG) mechanisms.
 - ▶ It shows that **DSIC welfare maximization** is possible in principle in **every multi-parameter environments**.

Outline

General Mechanism Design Environments

The VCG Mechanism

Remarks

Practical Implementation of Combinatorial Auctions
Indirect Mechanisms

Outline

General Mechanism Design Environments

The VCG Mechanism

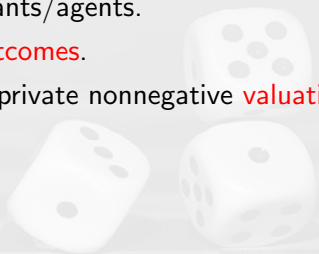
Remarks

Practical Implementation of Combinatorial Auctions
Indirect Mechanisms



General Multi-Parameter Design Environment

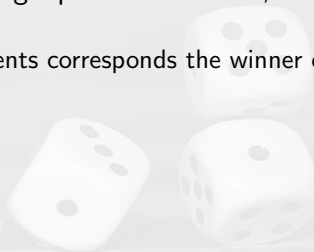
- ▶ n strategic participants/agents.
- ▶ a finite set Ω of **outcomes**.
- ▶ each agent i has a private nonnegative **valuation** $v_i(\omega)$ **for each outcome** $\omega \in \Omega$.



Single-Item Auction Revisited

Consider the single-item auction.

- ▶ In a the standard single parameter model,
 - ▶ $|\Omega| = n + 1$.
 - ▶ The $n + 1$ elements corresponds the winner of the item (if any).



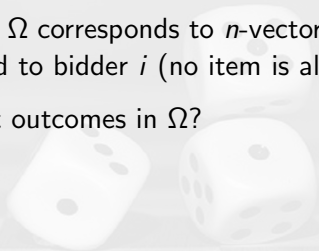
Single-Item Auction Revisited

Consider the single-item auction.

- ▶ In a the standard single parameter model,
 - ▶ $|\Omega| = n + 1$.
 - ▶ The $n + 1$ elements corresponds the winner of the item (if any).
 - ▶ The valuation of a bidder is 0 in all of the n outcomes (in which she doesn't win), leaving only one unknown parameter per bidder.
- ▶ In the general multi-parameter framework, a bidder can have a different valuation for each possible outcome (i.e., winner; competitor).

Combinatorial Auctions

- ▶ Multiple indivisible items are for sale.
- ▶ We have n bidders and a set M of m items.
- ▶ Bidders can have **preferences** between different **subsets** (bundles) of items.
- ▶ The outcome space Ω corresponds to n -vectors (S_1, \dots, S_n) , where $S_i \subseteq M$ is the bundle allocated to bidder i (no item is allocated twice).
- ★ How many different outcomes in Ω ?



Combinatorial Auctions

- ▶ Multiple indivisible items are for sale.
- ▶ We have n bidders and a set M of m items.
- ▶ Bidders can have **preferences** between different **subsets** (bundles) of items.
- ▶ The outcome space Ω corresponds to n -vectors (S_1, \dots, S_n) , where $S_i \subseteq M$ is the bundle allocated to bidder i (no item is allocated twice).
- ★ How many different outcomes in Ω ?
 - ▶ $(n + 1)^m$.

Combinatorial Auctions

- ▶ Multiple indivisible items are for sale.
- ▶ We have n bidders and a set M of m items.
- ▶ Bidders can have **preferences** between different **subsets** (bundles) of items.
- ▶ The outcome space Ω corresponds to n -vectors (S_1, \dots, S_n) , where $S_i \subseteq M$ is the bundle allocated to bidder i (no item is allocated twice).
- ★ How many different outcomes in Ω ?
 - ▶ $(n + 1)^m$.
- ▶ Each bidder has a private valuation $v_i(S)$ for each $S \subseteq M$.
- ★ So, how many private parameters does each bidder have?

Combinatorial Auctions

- ▶ Multiple indivisible items are for sale.
- ▶ We have n bidders and a set M of m items.
- ▶ Bidders can have **preferences** between different **subsets** (bundles) of items.
- ▶ The outcome space Ω corresponds to n -vectors (S_1, \dots, S_n) , where $S_i \subseteq M$ is the bundle allocated to bidder i (no item is allocated twice).
- ★ How many different outcomes in Ω ?
 - ▶ $(n + 1)^m$.
- ▶ Each bidder has a private valuation $v_i(S)$ for each $S \subseteq M$.
- ★ So, how many private parameters does each bidder have?
 - ▶ 2^m .

Outline

General Mechanism Design Environments

The VCG Mechanism

Remarks

Practical Implementation of Combinatorial Auctions
Indirect Mechanisms



The General Theorem

Theorem 7.3 (Multi-Parameter Welfare Maximization)

In every general mechanism design environment, there is a DSIC welfare-maximizing mechanism.

- ▶ Note that the computational efficiency is not asserted here.

The General Theorem

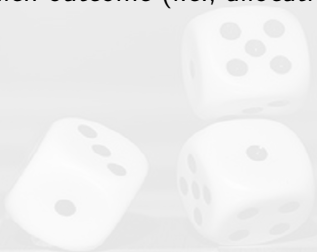
Theorem 7.3 (Multi-Parameter Welfare Maximization)

In every general mechanism design environment, there is a DSIC welfare-maximizing mechanism.

- ▶ Note that the computational efficiency is not asserted here.
- ▶ Let's discuss the main ideas behind the theorem before proving it.

Consider the Two-Step Approach as Usual

- ▶ First, assume that the agents truthfully report their private information (i.e., $\mathbf{b} = \mathbf{v}$).
- ▶ Then, figure out which outcome (i.e., allocation) to pick.



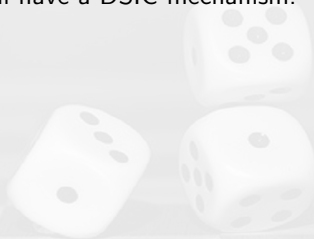
Consider the Two-Step Approach as Usual

- ▶ First, assume that the agents truthfully report their private information (i.e., $\mathbf{b} = \mathbf{v}$).
- ▶ Then, figure out which outcome (i.e., allocation) to pick.
- ▶ Pick a welfare-maximizing outcome using bids as proxies for the unknown valuations.
- ▶ We define the allocation rule

$$\omega^* := \mathbf{x}(\mathbf{b}) = \arg \max_{\omega \in \Omega} \sum_{i=1}^n b_i(\omega).$$

The Second Step

- ▶ Define the payment rule to incentivize the agents!
 - ▶ Hopefully we will have a DSIC mechanism.



The Second Step

- ▶ Define the payment rule to incentivize the agents!
 - ▶ Hopefully we will have a DSIC mechanism.
- ▶ However, unlike the single-parameter environments, here:
 - ▶ The report is multi-dimensional.
 - ▶ Myerson's lemma seems not to hold beyond single-parameter environments.
 - ▶ It's unclear how to define "monotonicity", "critical bid", etc.

A Key Idea to the Payment Rule

- ▶ Charge the agent the “externality” caused by agent i .
 - ▶ The welfare loss inflicted on the other $n - 1$ agents by agent i 's presence.
- ▶ This remains well defined in general mechanism design environments!
- ▶ The corresponding payment rule:

$$p_i(\mathbf{b}) = \left(\max_{\omega \in \Omega} \sum_{j \neq i} b_j(\omega) \right) - \sum_{j \neq i} b_j(\omega^*),$$

A Key Idea to the Payment Rule

- ▶ Charge the agent the “externality” caused by agent i .
 - ▶ The welfare loss inflicted on the other $n - 1$ agents by agent i 's presence.
- ▶ This remains well defined in general mechanism design environments!
- ▶ The corresponding payment rule:

$$p_i(\mathbf{b}) = \left(\max_{\omega \in \Omega} \sum_{j \neq i} b_j(\omega) \right) - \sum_{j \neq i} b_j(\omega^*),$$

- ▶ You may check that the payment $p_i(\mathbf{b}) \geq 0$.

The VCG Mechanism

Definition: VCG Mechanism

A mechanism (\mathbf{x}, \mathbf{p}) with allocation and payment rule as

$$\omega^* := \mathbf{x}(\mathbf{b}) = \arg \max_{\omega \in \Omega} \sum_{i=1}^n b_i(\omega).$$

and

$$p_i(\mathbf{b}) = \left(\max_{\omega \in \Omega} \sum_{j \neq i} b_j(\omega) \right) - \sum_{j \neq i} b_j(\omega^*),$$

respectively, is a Vickrey-Clarke-Groves or mechanism.

Another Form of the Payment

$$p_i(\mathbf{b}) = \underbrace{b_i(\omega^*)}_{\text{bid}} - \underbrace{\left[\sum_{j=1}^n b_j(\omega^*) - \max_{\omega \in \Omega} \sum_{j \neq i} b_j(\omega) \right]}_{\text{rebate}}$$

- ▶ Imagine the “rebate” as the increase in welfare attributable to agent i 's presence.

Another Form of the Payment

$$p_i(\mathbf{b}) = \underbrace{b_i(\omega^*)}_{\text{bid}} - \underbrace{\left[\sum_{j=1}^n b_j(\omega^*) - \max_{\omega \in \Omega} \sum_{j \neq i} b_j(\omega) \right]}_{\text{rebate}}$$

- ▶ Imagine the “rebate” as the increase in welfare attributable to agent i 's presence.
- ▶ For example, in the second-price auction, say we have two agents with bids $b_1 > b_2$.

Another Form of the Payment

$$p_i(\mathbf{b}) = \underbrace{b_i(\omega^*)}_{\text{bid}} - \underbrace{\left[\sum_{j=1}^n b_j(\omega^*) - \max_{\omega \in \Omega} \sum_{j \neq i} b_j(\omega) \right]}_{\text{rebate}}$$

- ▶ Imagine the “rebate” as the increase in welfare attributable to agent i 's presence.
- ▶ For example, in the second-price auction, say we have two agents with bids $b_1 > b_2$.
 - ▶ The highest bidder pays $b_1 - (b_1 - b_2) = b_2$!

Another Form of the Payment

$$p_i(\mathbf{b}) = \underbrace{b_i(\omega^*)}_{\text{bid}} - \underbrace{\left[\sum_{j=1}^n b_j(\omega^*) - \max_{\omega \in \Omega} \sum_{j \neq i} b_j(\omega) \right]}_{\text{rebate}} \leq b_i(\omega^*).$$

- ▶ Imagine the “rebate” as the increase in welfare attributable to agent i 's presence.
- ▶ For example, in the second-price auction, say we have two agents with bids $b_1 > b_2$.
 - ▶ The highest bidder pays $b_1 - (b_1 - b_2) = b_2$!
- ▶ Truthful reporting always guarantees nonnegative utility!

Proof of Theorem 7.3 (1/2)

- ▶ Fix an arbitrary general mechanism design environment.
- ▶ Let (\mathbf{x}, \mathbf{p}) denote the corresponding CVG mechanism.
- ▶ The mechanism maximizes the social welfare whenever the reports are truthful or not (by definition).
- ▶ Next, we have to verify the DSIC condition.
 - ▶ We need to show that for every agent i and every set \mathbf{b}_{-i} , agent i maximizes her utility $v_i(\mathbf{x}(\mathbf{b})) - p_i(\mathbf{b})$ by setting $\mathbf{b}_i = \mathbf{v}_i$.

Proof of Theorem 7.3 (2/2)

- Fix i and \mathbf{b}_{-i} . When the chosen outcome $\mathbf{x}(\mathbf{b})$ is ω^* , we have

$$v_i(\omega^*) - p_i(\mathbf{b}) = \underbrace{\left[v_i(\omega^*) + \sum_{j \neq i} b_j(\omega^*) \right]}_{(A)} - \underbrace{\left[\max_{\omega \in \Omega} \sum_{j \neq i} b_j(\omega) \right]}_{(B)}.$$

Proof of Theorem 7.3 (2/2)

- ▶ Fix i and \mathbf{b}_{-i} . When the chosen outcome $\mathbf{x}(\mathbf{b})$ is ω^* , we have

$$v_i(\omega^*) - p_i(\mathbf{b}) = \underbrace{\left[v_i(\omega^*) + \sum_{j \neq i} b_j(\omega^*) \right]}_{(A)} - \underbrace{\left[\max_{\omega \in \Omega} \sum_{j \neq i} b_j(\omega) \right]}_{(B)}.$$

- ▶ (B) can be viewed as a constant. So we focus on maximizing (A).

Proof of Theorem 7.3 (2/2)

- ▶ Fix i and \mathbf{b}_{-i} . When the chosen outcome $\mathbf{x}(\mathbf{b})$ is ω^* , we have

$$v_i(\omega^*) - p_i(\mathbf{b}) = \underbrace{\left[v_i(\omega^*) + \sum_{j \neq i} b_j(\omega^*) \right]}_{(A)} - \underbrace{\left[\max_{\omega \in \Omega} \sum_{j \neq i} b_j(\omega) \right]}_{(B)}.$$

- ▶ (B) can be viewed as a constant. So we focus on maximizing (A).
- ▶ For agent i , assume that ω^* is chosen, setting $b_i = v_i$ makes (A) maximized.
 - ▶ $v_i(\omega^*) + \sum_{j \neq i} b_j(\omega^*) = \sum_j b_j(\omega^*)$.
 - ▶ Recall that $\omega^* = \arg \max_{\omega \in \Omega} \sum_{i=1}^n b_i(\omega)$.

Outline

General Mechanism Design Environments

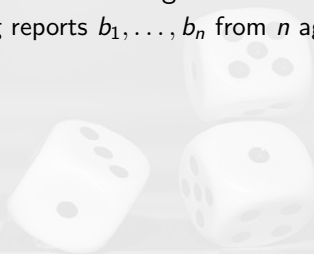
The VCG Mechanism

Remarks

Practical Implementation of Combinatorial Auctions
Indirect Mechanisms

Practical Considerations (1/3)

- ▶ Preference elicitation is a challenge.
 - ▶ Consider getting reports b_1, \dots, b_n from n agents in a combinatorial auction with m items.



Practical Considerations (1/3)

- ▶ Preference elicitation is a challenge.
 - ▶ Consider getting reports b_1, \dots, b_n from n agents in a combinatorial auction with m items.
 - ▶ Each bidder has 2^m private parameters!
 - ▶ It's hard for her to figure out or write down so many numbers.
 - ▶ No seller would want to read them.

Practical Considerations (2/3)

- ▶ As in single-parameter environments, welfare maximization could be a computationally intractable.
 - ▶ Recall the knapsack auction.
- ▶ Sometimes even approximate welfare maximization is still computationally intractable.

Practical Considerations (3/3)

- ▶ VCG mechanisms can have bad revenue.

Exercise 4 (5%)

1. Consider a combinatorial auction with two bidders and two items A and B . The first bidder only wants both items, so $v_1(\{A, B\}) = 1$ and is 0 otherwise. The second bidder only wants item A , so $v_2(\{A, B\}) = v_2(\{A\}) = 1$ and is 0 otherwise. **Please show that the revenue of the VCG mechanism is 1 in this example.**
2. Now suppose that we add a third bidder who **only wants item B** , so $v_3(\{A, B\}) = v_3(\{B\}) = 1$. **Please show that the maximum welfare is 2 but the VCG revenue is 0 in this case.**

Recall the issues

- ▶ Combinatorial auction: n bidders, m items, bidder i 's valuation $v_i(S)$ for each bundle S of items.
- ▶ The number of parameters that each bidder reports in the VCG mechanism (or any other direct-revelation mechanism) grows **exponentially** with m .

Indirect Mechanism

- ▶ Learn information about bidders' preferences only on a **need-to-know** basis.
- ▶ The canonical indirect auction: **ascending English auction**.
 - ★ An auctioneer asks for takers at successively higher prices.
 - ★ The auction ends when no one accepts the currently proposed price.
 - ★ The winner (if any): the bidder who accepted the previously proposed price
 - ✓ This previous price is the final sale price.

Indirect Mechanism

- ▶ Learn information about bidders' preferences only on a **need-to-know** basis.
- ▶ The canonical indirect auction: **ascending English auction**.
 - ★ An auctioneer asks for takers at successively higher prices.
 - ★ The auction ends when no one accepts the currently proposed price.
 - ★ The winner (if any): the bidder who accepted the previously proposed price
 - ✓ This previous price is the final sale price.
- ▶ Empirically, bidders are more likely to play their dominant strategies in this kind of auction than a sealed-bid second-price auction.
 - ▶ Bidders: not likely to overbid;
 - ▶ Seller: only learns a lower bound on the highest bid.

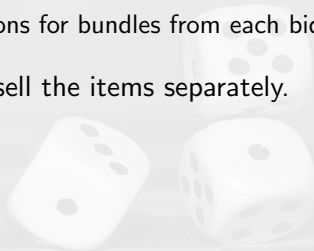
How about selling items separately?

- ▶ So what's a natural indirect auction format for combinatorial auctions?
 - ▶ Eliciting valuations for bundles from each bidder is avoided.



How about selling items separately?

- ▶ So what's a natural indirect auction format for combinatorial auctions?
 - ▶ Eliciting valuations for bundles from each bidder is avoided.
- ▶ The simplest way: sell the items separately.



How about selling items separately?

- ▶ So what's a natural indirect auction format for combinatorial auctions?
 - ▶ Eliciting valuations for bundles from each bidder is avoided.
- ▶ The simplest way: sell the items separately.
- ▶ What's the issue or problem?

Substitutes vs. Complements

For two items A and B ,

- ▶ **substitute condition:** $v(AB) \leq v(A) + v(B)$.
 - ▶ Spectrum auction: two licenses (the same area & equal-sized frequency ranges).
 - ▶ iPhone 13 + iPhone 14 announced together?
 - ★ Welfare maximization is computationally tractable.
- ▶ **complement condition:** $v(AB) > v(A) + v(B)$.
 - ▶ Spectrum auction: a collection of licenses that are adjacent (geographically or frequency ranges).
 - ▶ Two items with additional enhancement when they are both provided. [★]
Welfare maximization is computationally *intractable*.
- ▶ In real world: mixture of substitutes and complements.

Typical mistake #1 of running separate single-item auctions

Hold the single-item auctions **sequentially**, one at a time.

- ▶ The scenario: A sequence of single-item auctions for two identical items.
 - ▶ Items are sold via back-to-back second-price auctions.
- ▶ Let's say you are a bidder with a **VERY HIGH** valuation.
- ▶ Suppose that every other bidder bid truthfully.

Typical mistake #1 of running separate single-item auctions

Hold the single-item auctions **sequentially**, one at a time.

- ▶ The scenario: A sequence of single-item auctions for two identical items.
 - ▶ Items are sold via back-to-back second-price auctions.
- ▶ Let's say you are a bidder with a VERY HIGH valuation.
- ▶ Suppose that every other bidder bid truthfully.
- ★ If you participate in the first auction, you will win and pay the second-highest valuation.

Typical mistake #1 of running separate single-item auctions

Hold the single-item auctions **sequentially**, one at a time.

- ▶ The scenario: A sequence of single-item auctions for two identical items.
 - ▶ Items are sold via back-to-back second-price auctions.
- ▶ Let's say you are a bidder with a VERY HIGH valuation.
- ▶ Suppose that every other bidder bid truthfully.
- ★ If you participate in the first auction, you will win and pay the second-highest valuation.
- ★ If you skip it, the bidder with the second-highest valuation wins the first auction and disappear.
 - Then you would win the second auction at a price equal to the **third-highest** valuation.

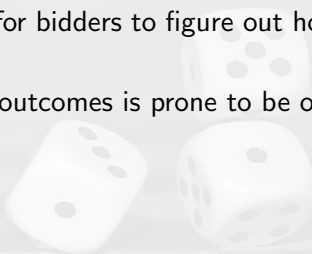
A Story from the textbook

- ▶ In March 2000, Switzerland auctioned off three blocks of spectrum via a **sequence of second-price auctions**.
- ▶ The first two auctions were for identical items, 28 MHz blocks, and sold for 121 million and 134 million Swiss francs, respectively.
 - ▶ This is already more price variation than one would like for identical items.
- ▶ But in the third auction, where a larger 56 MHz block was being sold, the selling price was only 55 million francs!
- ▶ Some of the bids must have been far from optimal, and both the welfare and revenue achieved by this auction are suspect.

Typical mistake #2 of running separate single-item auctions

Use sealed-bid single-item auctions (simultaneously).

- ▶ Again, it's difficult for bidders to figure out how to bid, especially for multiple items.
- ▶ The challenge: the outcomes is prone to be of low welfare & revenue.



Typical mistake #2 of running separate single-item auctions

Use sealed-bid single-item auctions (simultaneously).

- ▶ Again, it's difficult for bidders to figure out how to bid, especially for multiple items.
- ▶ The challenge: the outcomes is prone to be of low welfare & revenue.
- ▶ Consider that there are 3 bidders and 2 identical items, and each bidder wants **only one**.
- ▶ With simultaneous second-price single-item auctions,

Typical mistake #2 of running separate single-item auctions

Use sealed-bid single-item auctions (simultaneously).

- ▶ Again, it's difficult for bidders to figure out how to bid, especially for multiple items.
- ▶ The challenge: the outcomes is prone to be of low welfare & revenue.
- ▶ Consider that there are 3 bidders and 2 identical items, and each bidder wants **only one**.
- ▶ With simultaneous second-price single-item auctions,
 - ▶ if each bidder targets only one item, one of the licenses is likely to have only one bidder and will be given away for **free** or sold at the reserve price.

A Story from the textbook

- ▶ In 1990, the New Zealand government auctioned off essentially identical licenses for television broadcasting using **simultaneous (sealed-bid) second-price auctions**.
- ▶ The revenue in the 1990 New Zealand auction was only \$36 million, a paltry fraction of the projected \$250 million.
 - ▶ On one license, the high bid was \$100,000 while the second-highest bid (and selling price) was \$6!
 - ▶ On another, the high bid was \$7 million and the second-highest was \$5,000.

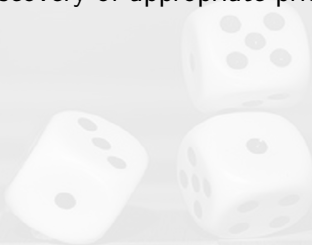
The prevailing simultaneous ascending auctions (SAAs)

- ▶ Conceptually, SAAs are like a bunch of single-item English auctions run in parallel in the same room and one auctioneer per item.



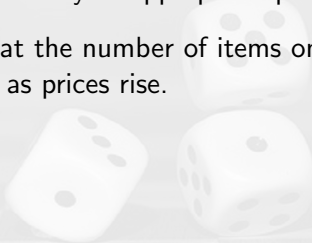
The prevailing simultaneous ascending auctions (SAAs)

- ▶ Conceptually, SAAs are like a bunch of single-item English auctions run in parallel in the same room and one auctioneer per item.
- ▶ All bidders have to participate in the auction from the beginning and contribute to the discovery of appropriate prices.



The prevailing simultaneous ascending auctions (SAAs)

- ▶ Conceptually, SAAs are like a bunch of single-item English auctions run in parallel in the same room and one auctioneer per item.
- ▶ All bidders have to participate in the auction from the beginning and contribute to the discovery of appropriate prices.
- ▶ The gist: require that the number of items on which a bidder bids only decreases over time as prices rise.



The prevailing simultaneous ascending auctions (SAAs)

- ▶ Conceptually, SAAs are like a bunch of single-item English auctions run in parallel in the same room and one auctioneer per item.
- ▶ All bidders have to participate in the auction from the beginning and contribute to the discovery of appropriate prices.
- ▶ The gist: require that the number of items on which a bidder bids only decreases over time as prices rise.
- ▶ The primary reason that SAAs work better is **price discovery**.
 - ▶ A bidder can abandon the items in the mid-course as she acquires better information about the likely selling prices of the items.

The prevailing simultaneous ascending auctions (SAAs)

- ▶ Conceptually, SAAs are like a bunch of single-item English auctions run in parallel in the same room and one auctioneer per item.
- ▶ All bidders have to participate in the auction from the beginning and contribute to the discovery of appropriate prices.
- ▶ The gist: require that the number of items on which a bidder bids only decreases over time as prices rise.
- ▶ The primary reason that SAAs work better is **price discovery**.
 - ▶ A bidder can abandon the items in the mid-course as she acquires better information about the likely selling prices of the items.
 - ▶ Again, bidder only need to determine their valuations on a **need-to-know** basis.
- ▶ Though such kind a mechanisms still has its vulnerabilities (skipped here for further readings).