Sequential Composition of Choice Functions

Sean Horan
Montreal

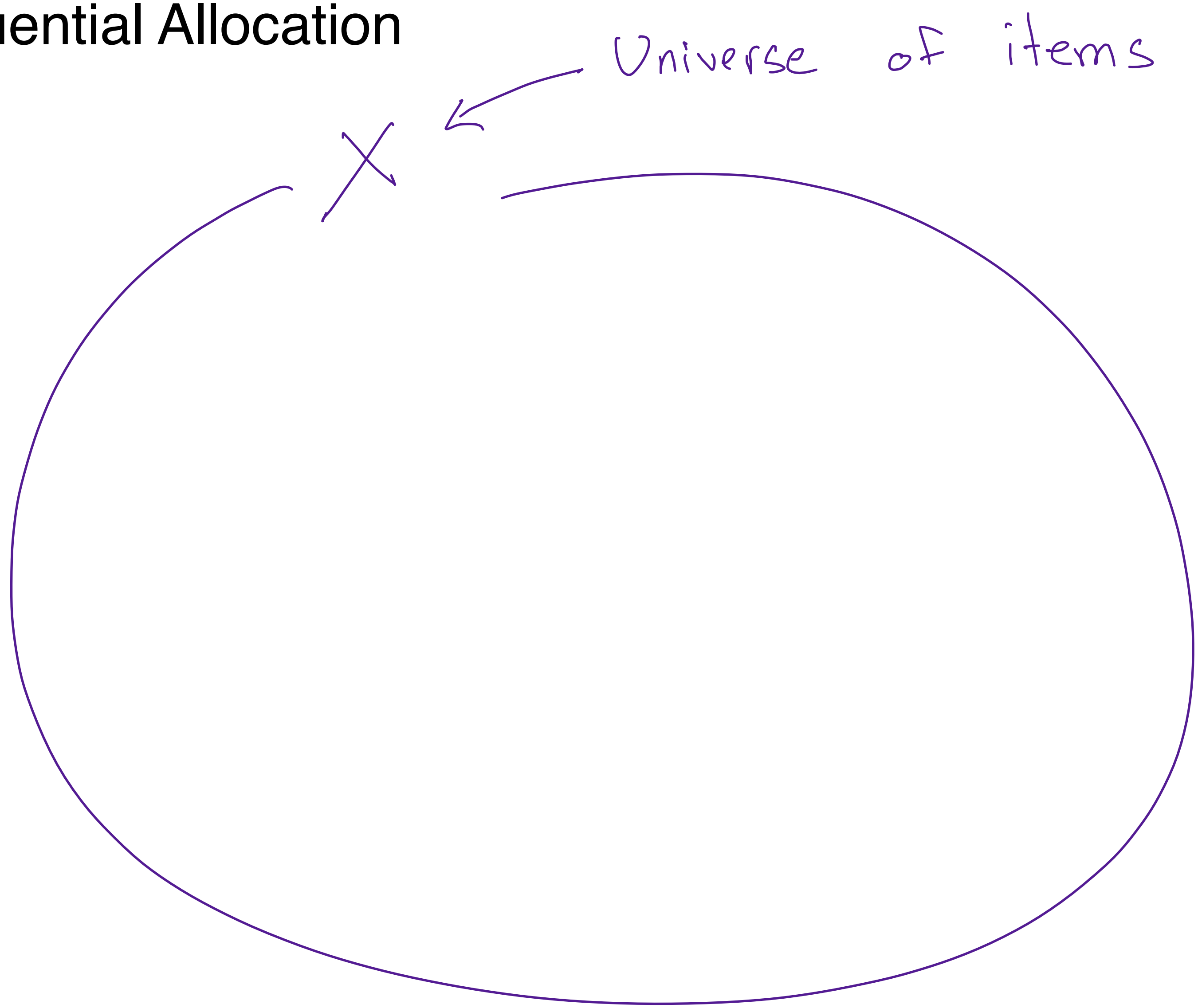
Vikram Manjunath
Ottawa

Conference on Economic Design

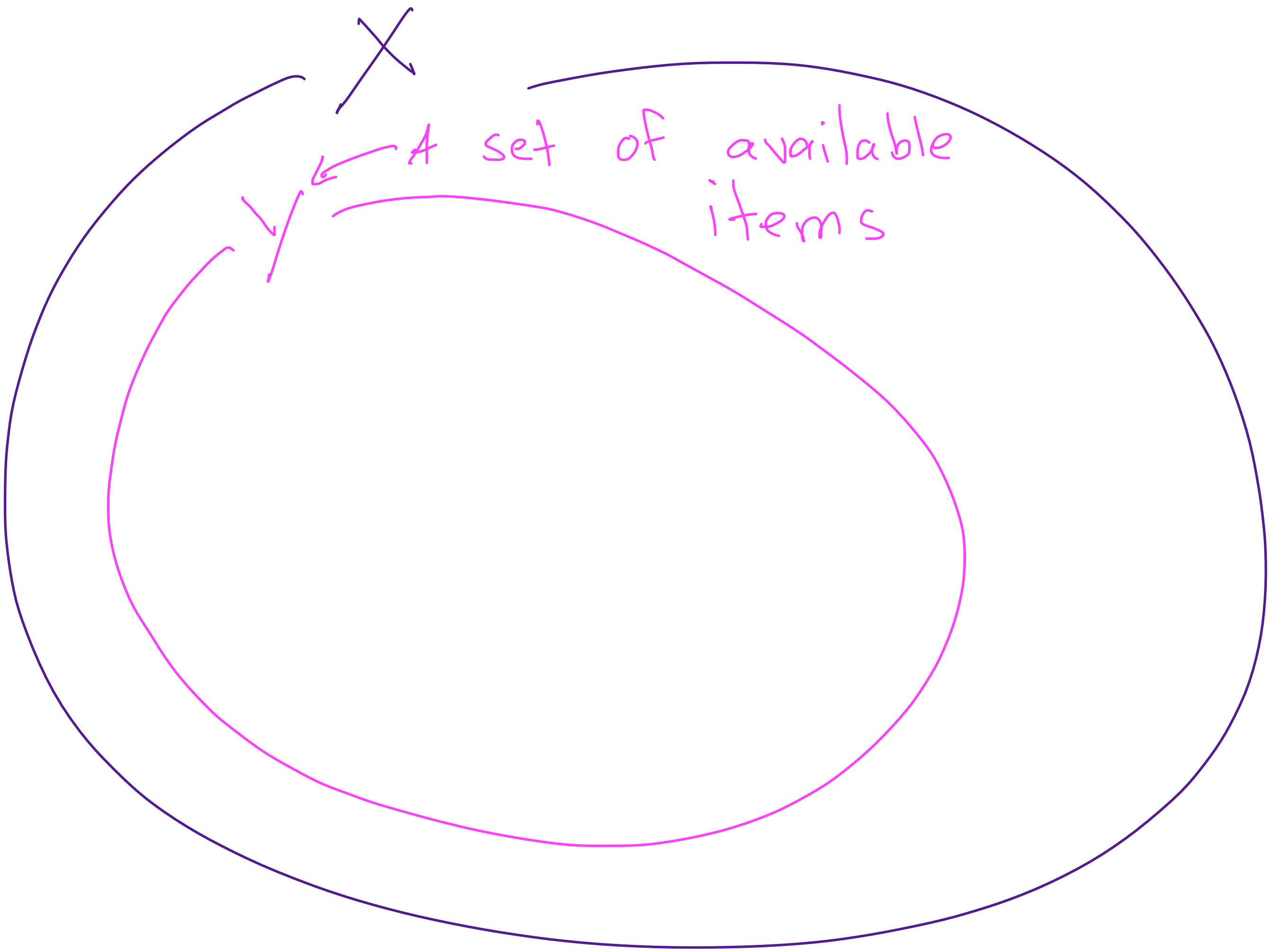
Girona

June 16, 2023

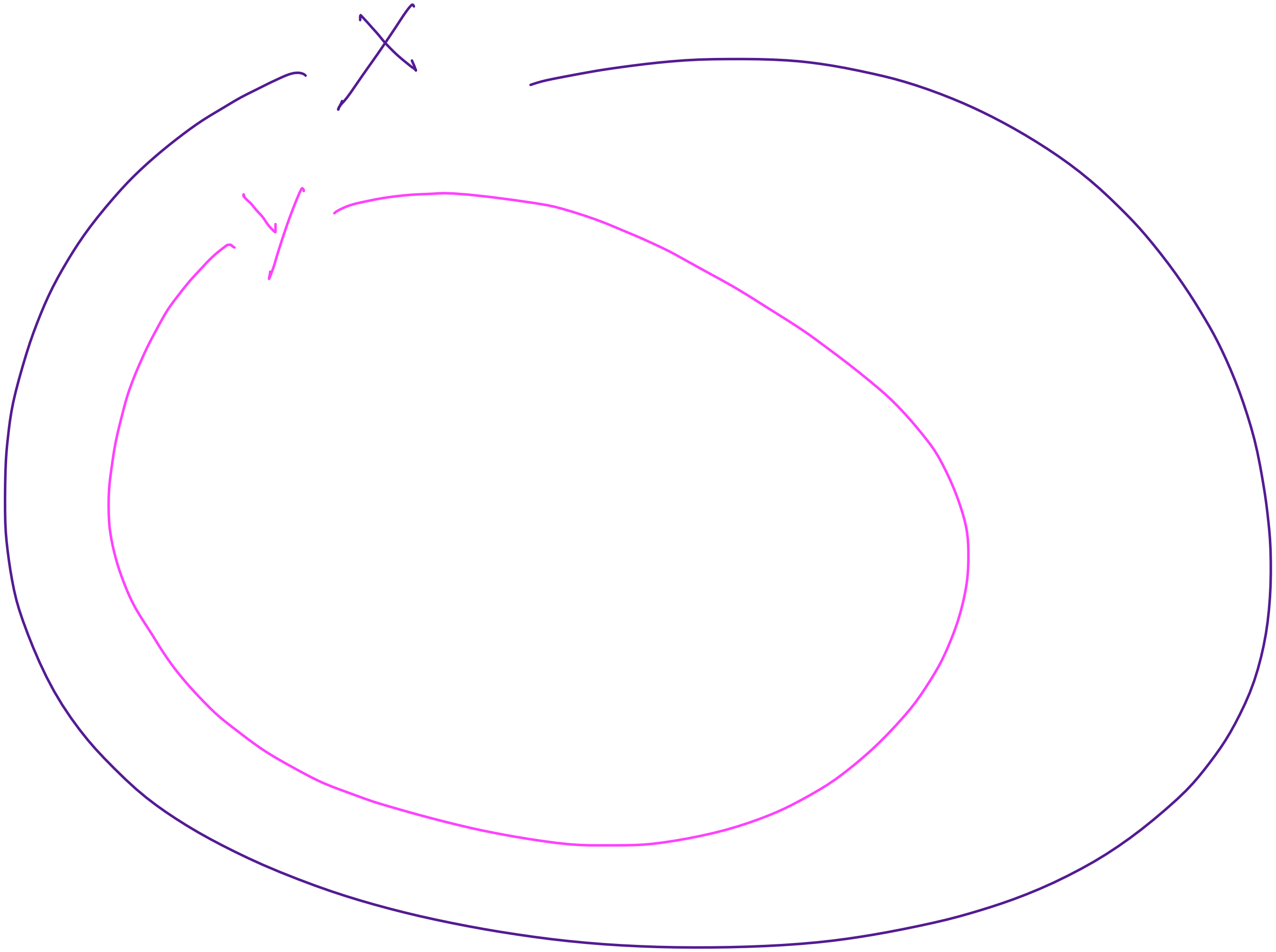
Sequential Allocation



Sequential Allocation

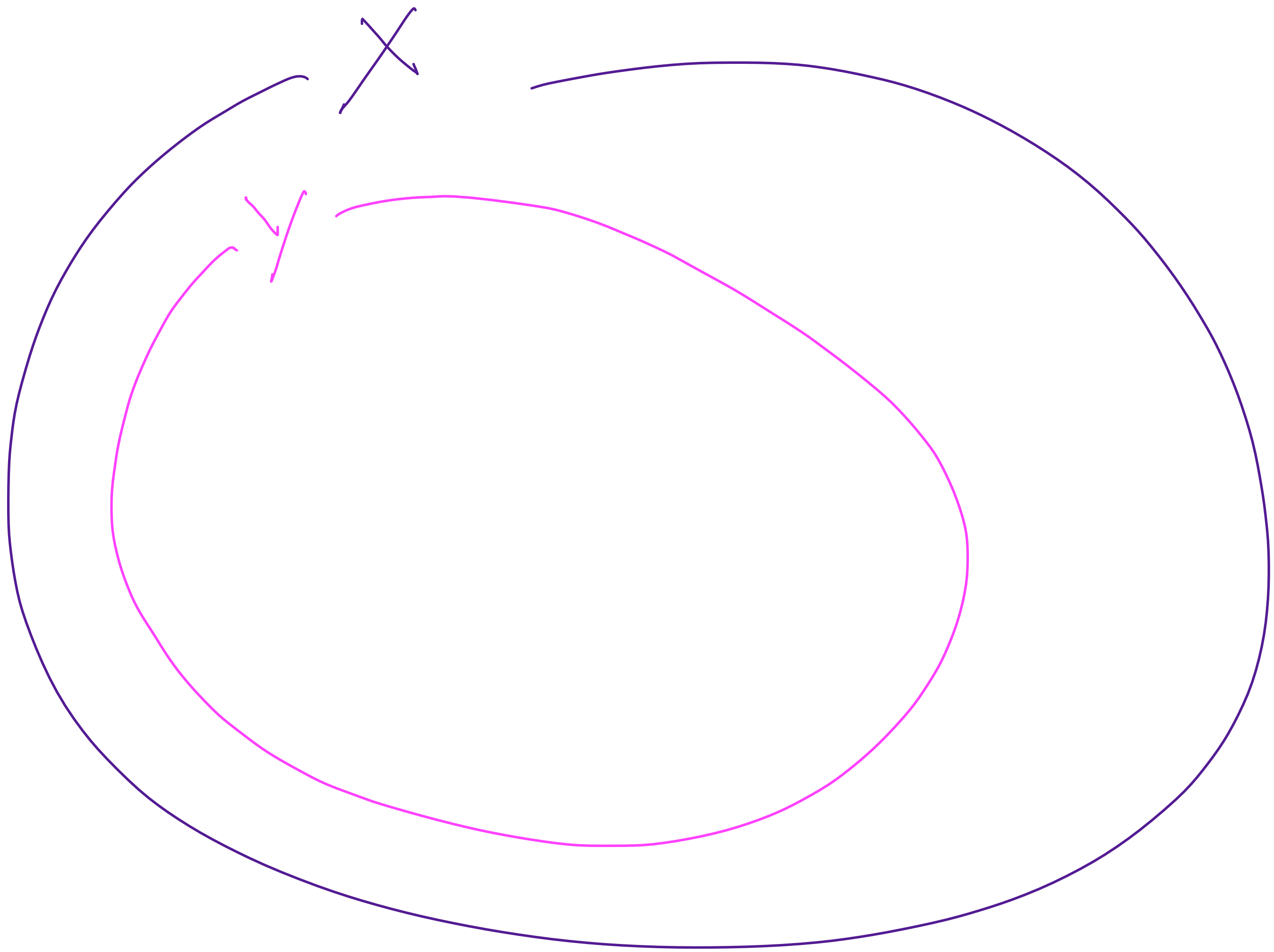


Sequential Composition



Agents: $i_1 \quad i_2 \quad i_3 \quad \ldots \quad i_n$

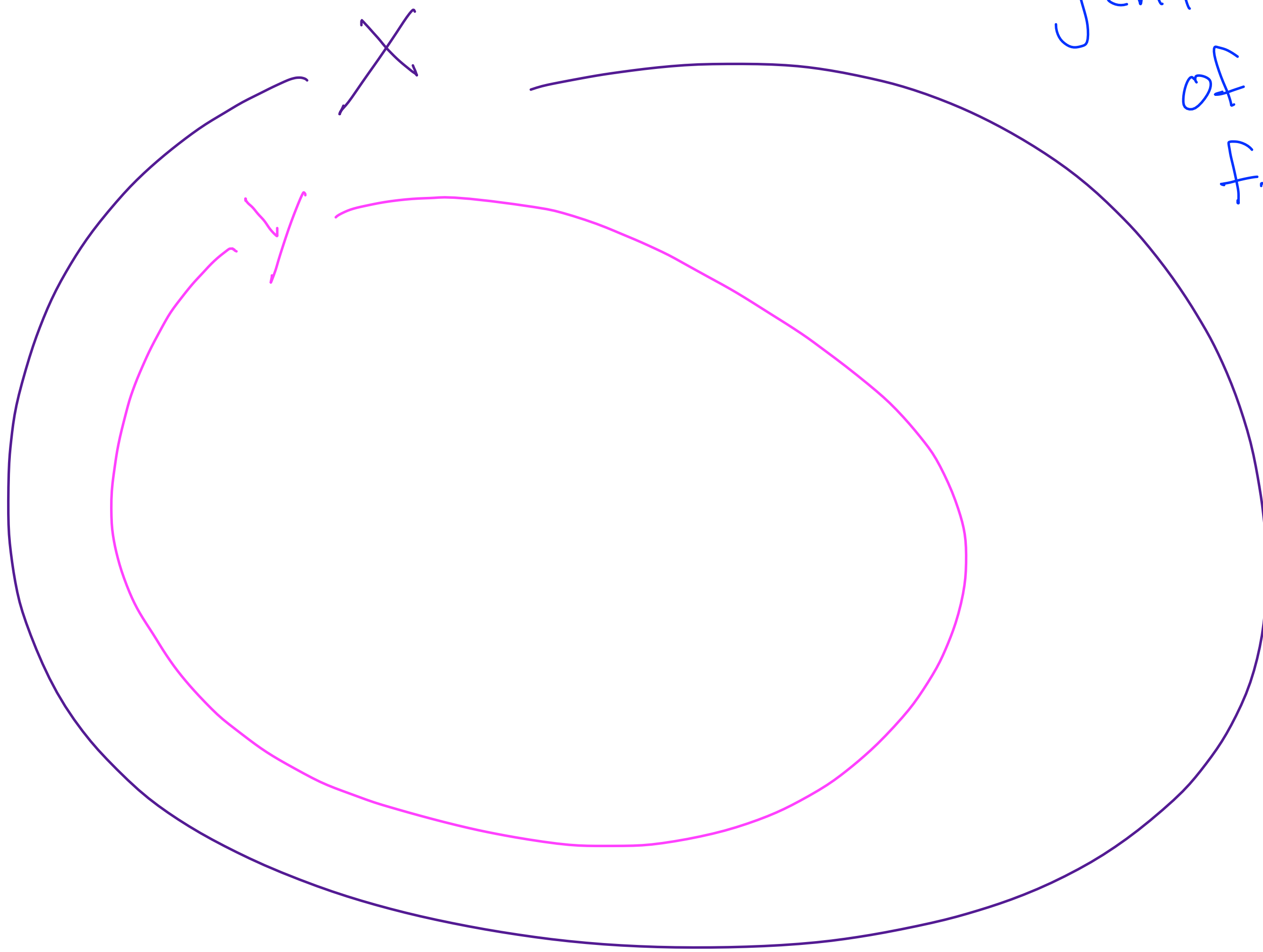
Sequential Composition



Agents: $i_1 \quad i_2 \quad i_3 \quad \ldots \quad i_n$

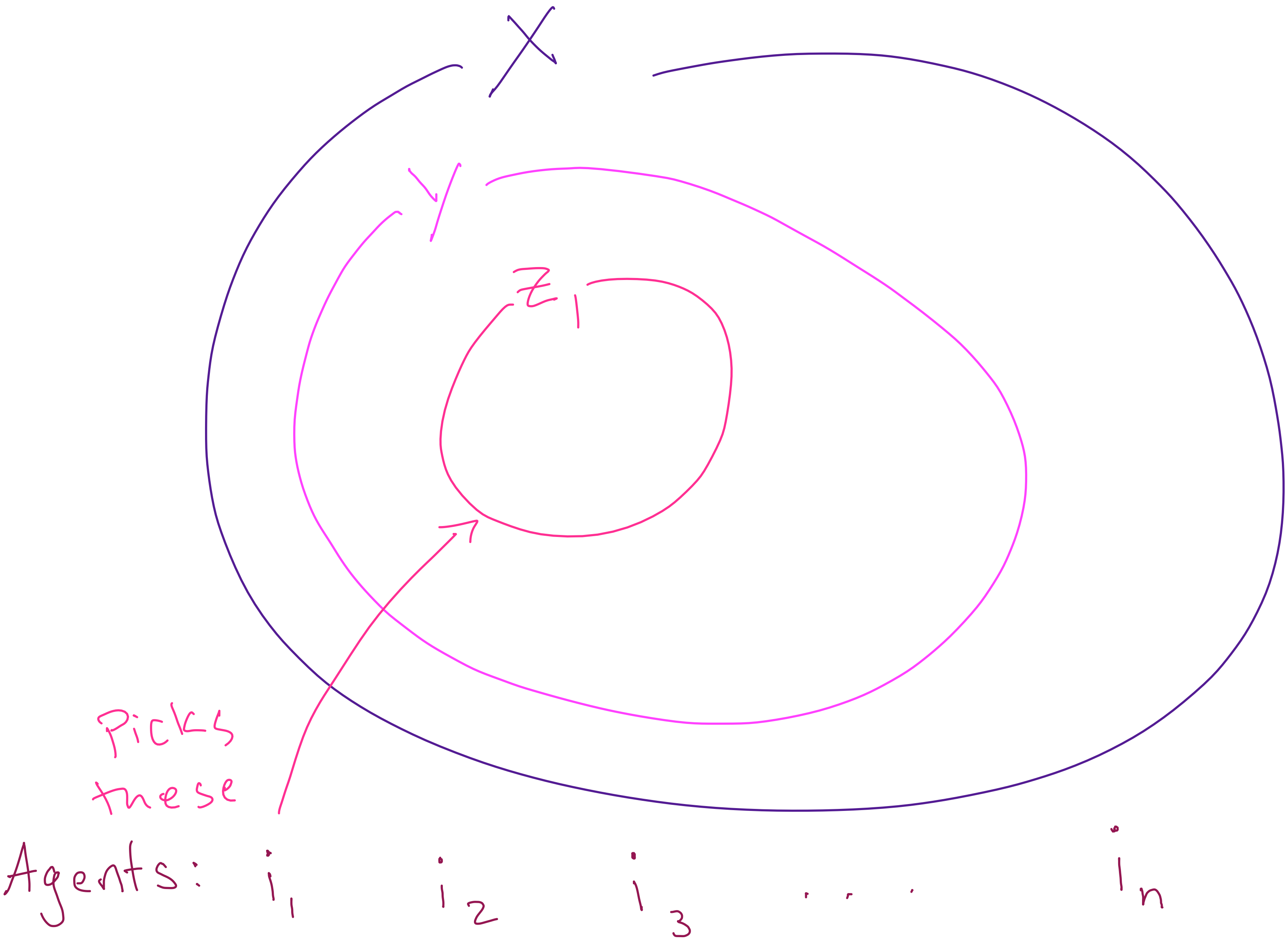
Have preferences over the subsets of items they could be assigned

Sequential Allocation ← A way to assign each agent a set of items from Y

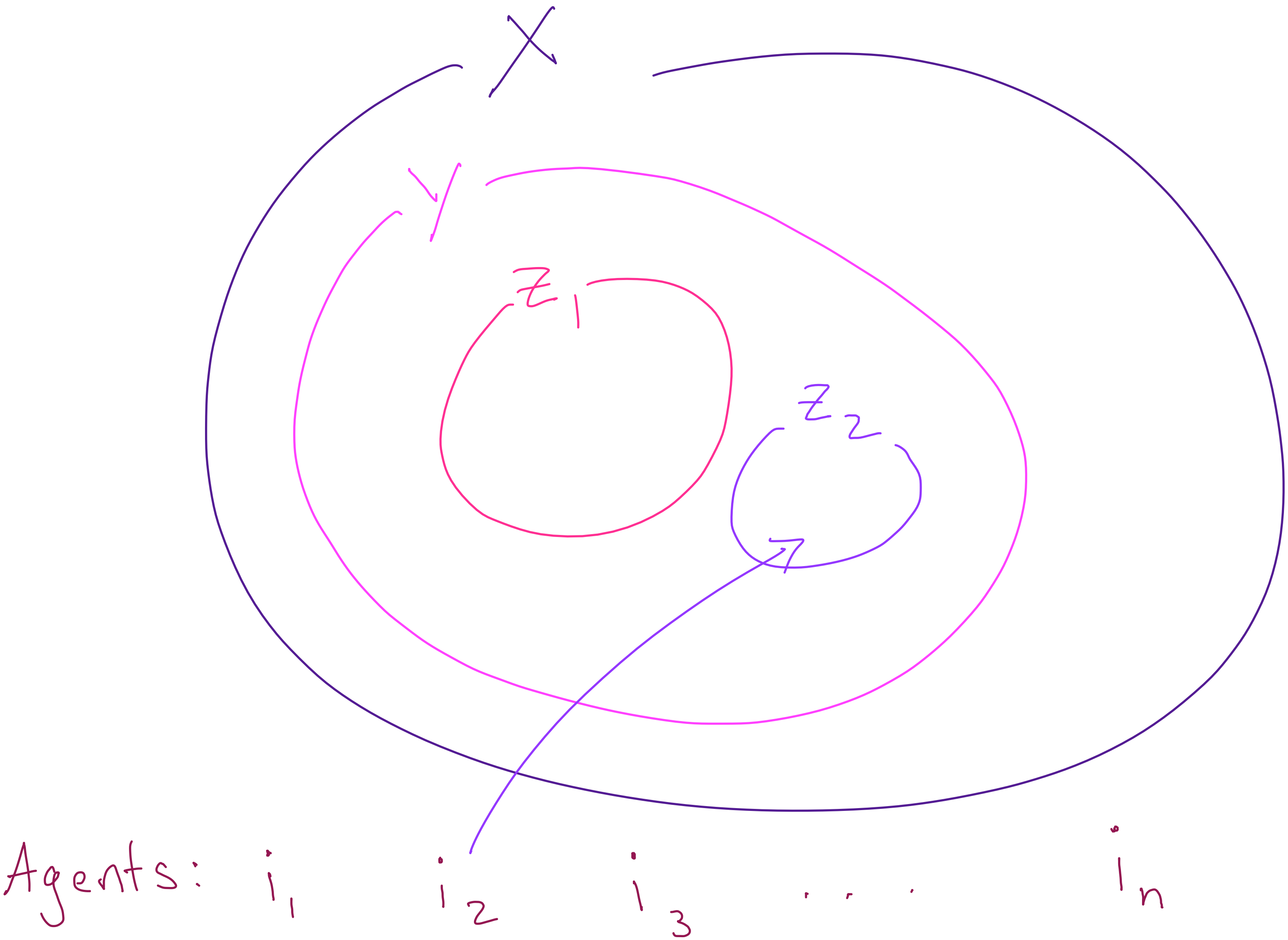


Agents: $i_1 \quad i_2 \quad i_3 \quad \ldots \quad i_n$

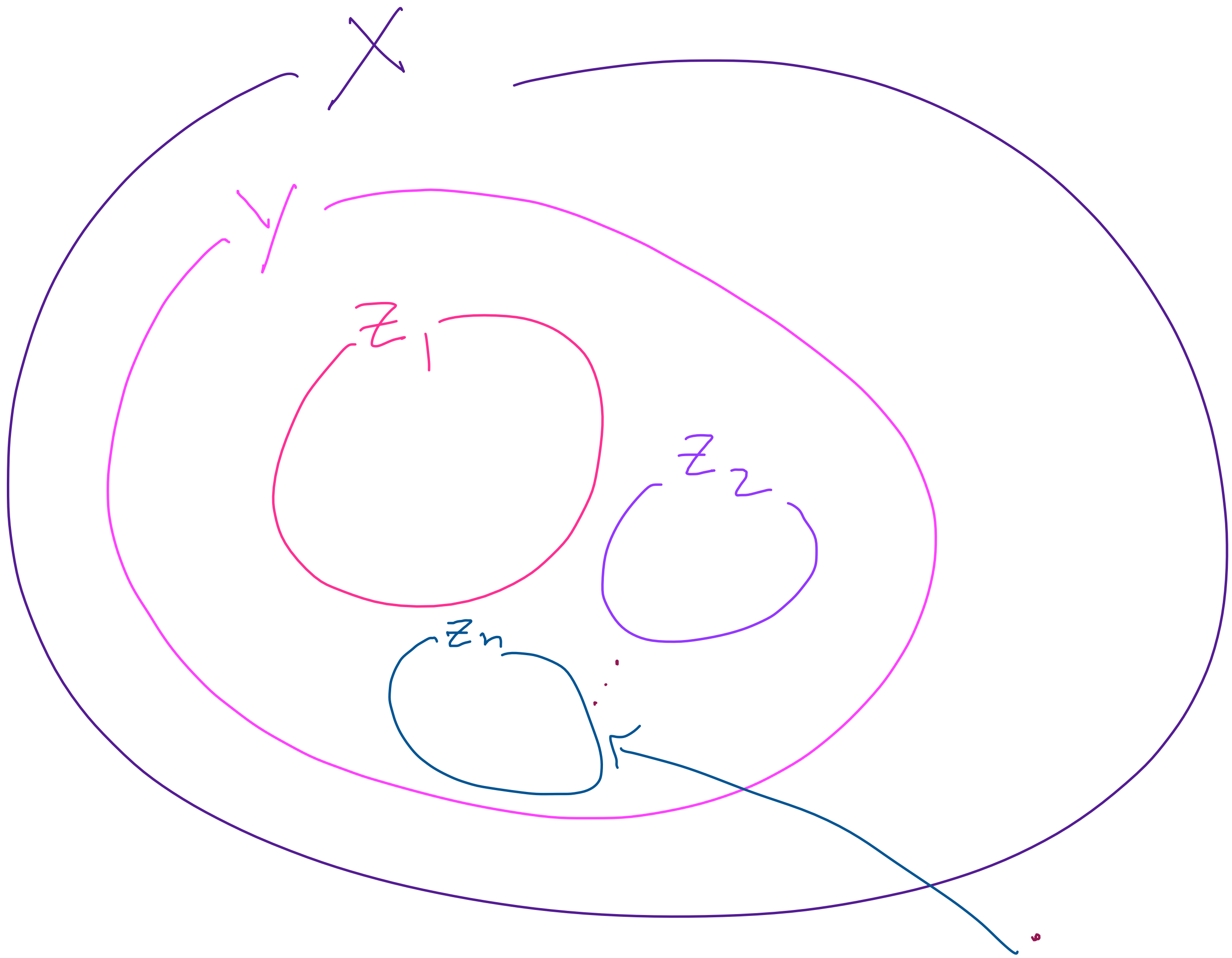
Sequential Allocation



Sequential Allocation

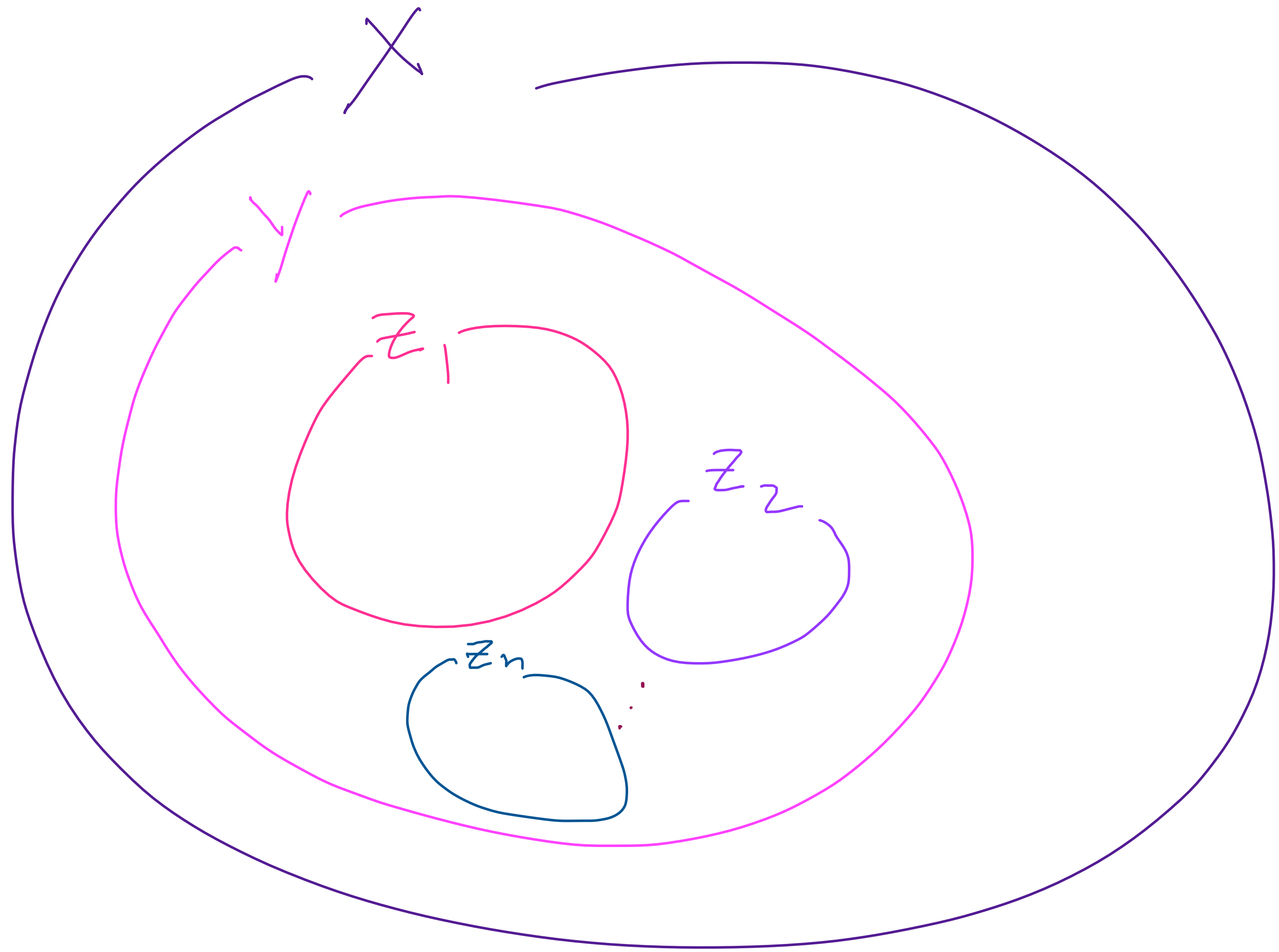


Sequential Allocation



Agents: i_1 i_2 i_3 . . . i_n

Sequential Allocation



Agents: $i_1 \quad i_2 \quad i_3 \quad \ldots \quad i_n$

Sequential Allocation

Why allocate sequentially?

Sequential Allocation

Why allocate sequentially?

– "Simple" allocation rule

Sequential Allocation

Why allocate sequentially?

- "Simple" allocation rule
  * Easy for participants to behave optimally (see e.g. Pycia & Troyan (forthcoming))
  * Computationally easy

Sequential Allocation

Why allocate sequentially?

Another (new?) perspective on designing allocation rules

Sequential Allocation

Why allocate sequentially?

Another (new?) perspective on designing allocation rules

Available items

+

Preferences of agents

Sequential Allocation

Why allocate sequentially?

Another (new?) perspective on designing allocation rules

Available items

+

Preferences of agents

Inputs

Sequential Allocation

Why allocate sequentially?

Another (new?) perspective on designing allocation rules

Available items + Preferences of agents $\longrightarrow$ Assignment of sets of items to agents

Inputs

Output

Sequential Allocation

Why allocate sequentially?

Another (new?) perspective on designing allocation rules

Available items

+

Preferences of agents

Inputs

$\longrightarrow$

Assignment of sets of items to agents

↓

Available items that are actually assigned

Output

Sequential Allocation

Why allocate sequentially?

Another (new?) perspective on designing allocation rules

Available items + Preferences of agents (Fix these) → Assignment of sets of items to agents → Available items that are actually assigned

Inputs

Output

Sequential Allocation

Why allocate sequentially?

Another (new?) perspective on designing allocation rules

Available items

Available items that are actually assigned

Inputs

Output

Sequential Allocation

Why allocate sequentially?

Another (new?) perspective on designing allocation rules

Available items

The allocation rule defines an "aggregated" choice function

Available items that are actually assigned

Inputs

Output

Sequential Allocation

Why allocate sequentially?

Another (new?) perspective on designing allocation rules

Available items

The allocation rule defines an "aggregated" choice function

(of course, this collapses the vector into a set)

Available items that are actually assigned

Inputs

Output

Sequential Allocation

Why allocate sequentially?

- Preserves "nice" properties of individual preferences

Sequential Allocation

Why allocate sequentially?

- Preserves "nice" properties of individual preferences

  If component choice functions are well behaved then their sequential composition is well behaved

Sequential Composition

Sequential Composition

$c_1, c_2, \ldots, c_n$ — n choice functions

reflecting agents' preferences

Sequential Composition

$c_1, c_2, \ldots, c_n$ – n choice functions

Y – A set to choose frome

Sequential Composition

$C_1, C_2, \ldots, C_n$ – n choice functions

Y – A set to choose from

$$Y \longrightarrow C_1$$

Sequential Composition

$c_1, c_2, \ldots, c_n$ – n choice functions

Y – A set to choose from

$$
\begin{tikzcd}
Y \arrow[d] & \\
c_1 \arrow[r, "\text{doesn't}", "\text{pick}"'] \arrow[d, "\text{picks}"'] & Y \setminus z_1 \\
z_1 &
\end{tikzcd}
$$

Sequential Composition

$c_1, c_2, \ldots, c_n$ – n choice functions

Y – A set to choose from

$$
\begin{tikzcd}
Y \arrow[d] & \\
c_1 \arrow[r, "\text{doesn't}", "\text{pick}"'] \arrow[d, "\text{picks}"'] & Y \setminus z_1 \arrow[d] \\
z_1 & c_2
\end{tikzcd}
$$

Sequential Composition

$c_1, c_2, \ldots, c_n$ – n choice functions

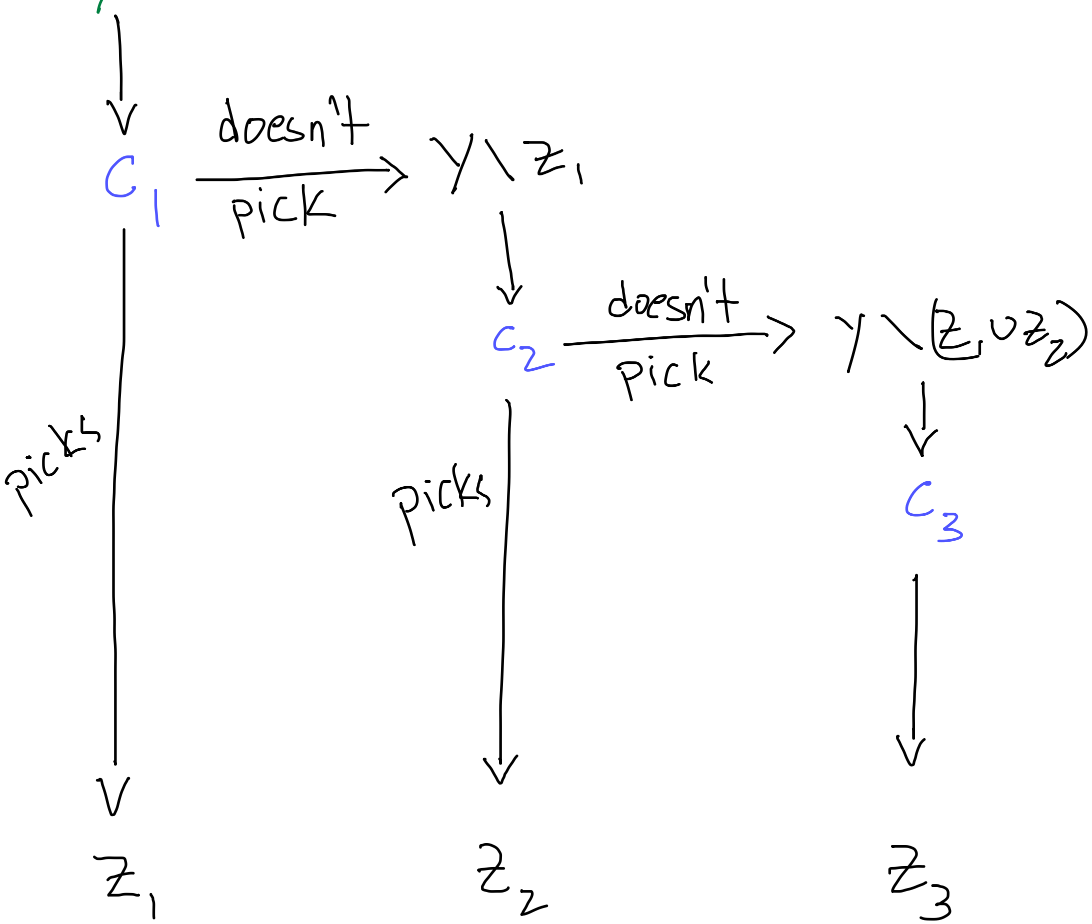
Y – A set to choose from

$$\begin{tikzcd}
Y \arrow[d] & \\
c_1 \arrow[r, "\text{doesn't}", "\text{pick}"'] \arrow[dd, "\text{picks}"'] & Y \setminus Z_1 \arrow[d] & \\
 & c_2 \arrow[r, "\text{doesn't}", "\text{pick}"'] \arrow[d, "\text{picks}"'] & Y \setminus (Z_1 \cup Z_2) \\
Z_1 & Z_2 &
\end{tikzcd}$$

Sequential Composition

$C_1, C_2, \ldots, C_n$ – n choice functions

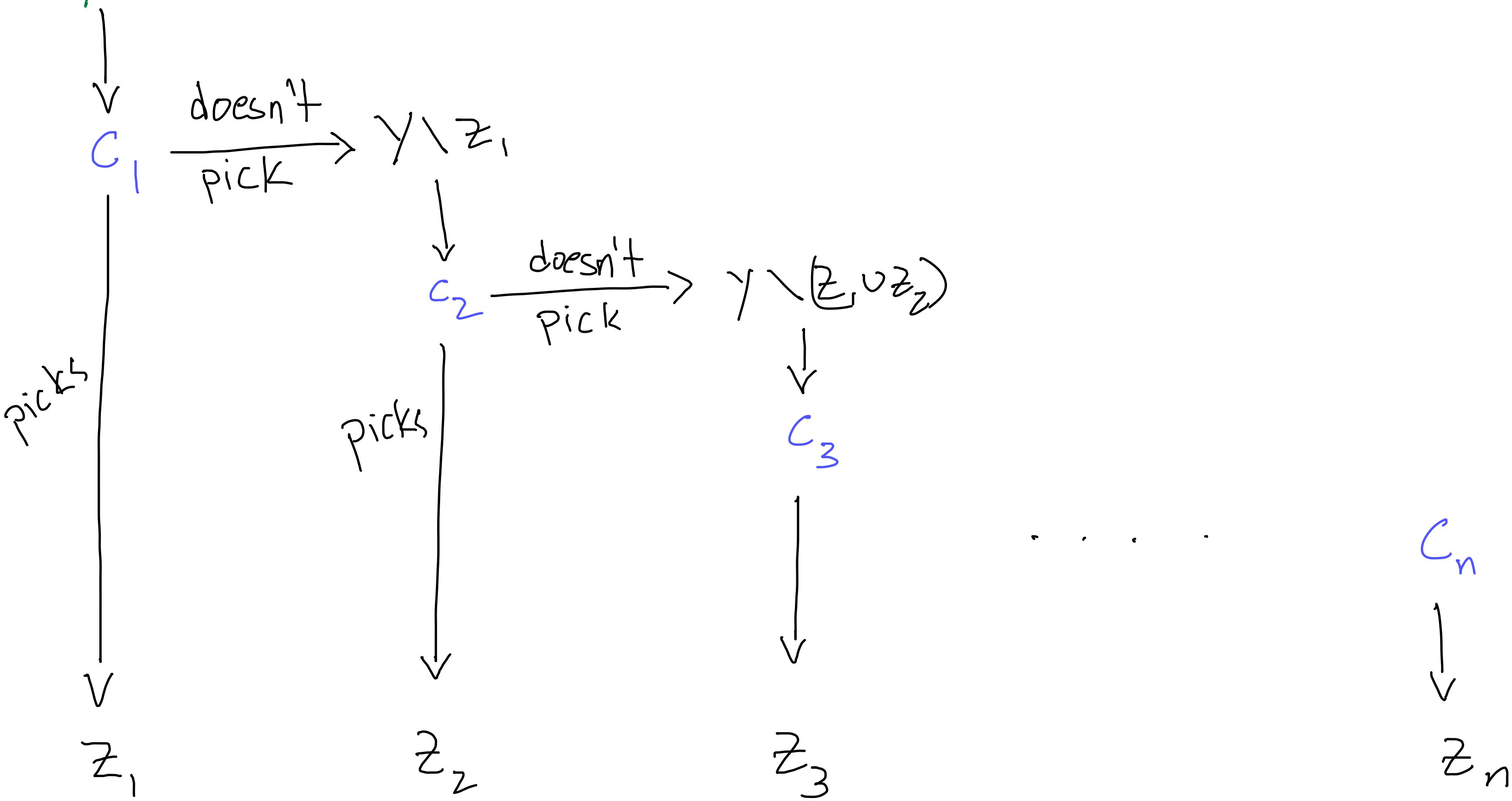
Y – A set to choose from



Sequential Composition

$c_1, c_2, \ldots, c_n$ – n choice functions

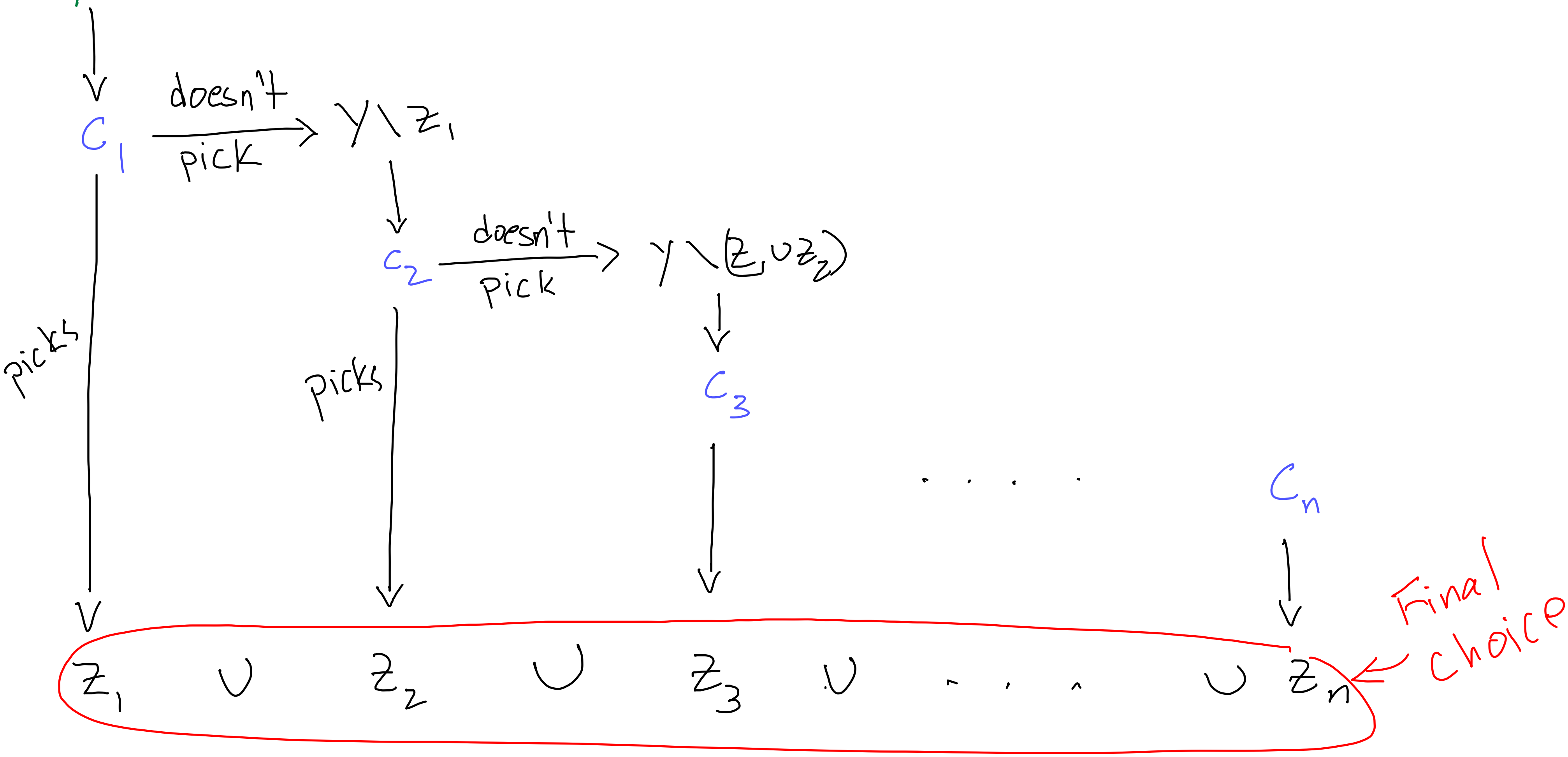
Y – A set to choose from



Sequential Composition

$c_1, c_2, \ldots, c_n$ – n choice functions

Y – A set to choose from



Sequential Composition

Example:

Organization has two divisions

Sequential Composition

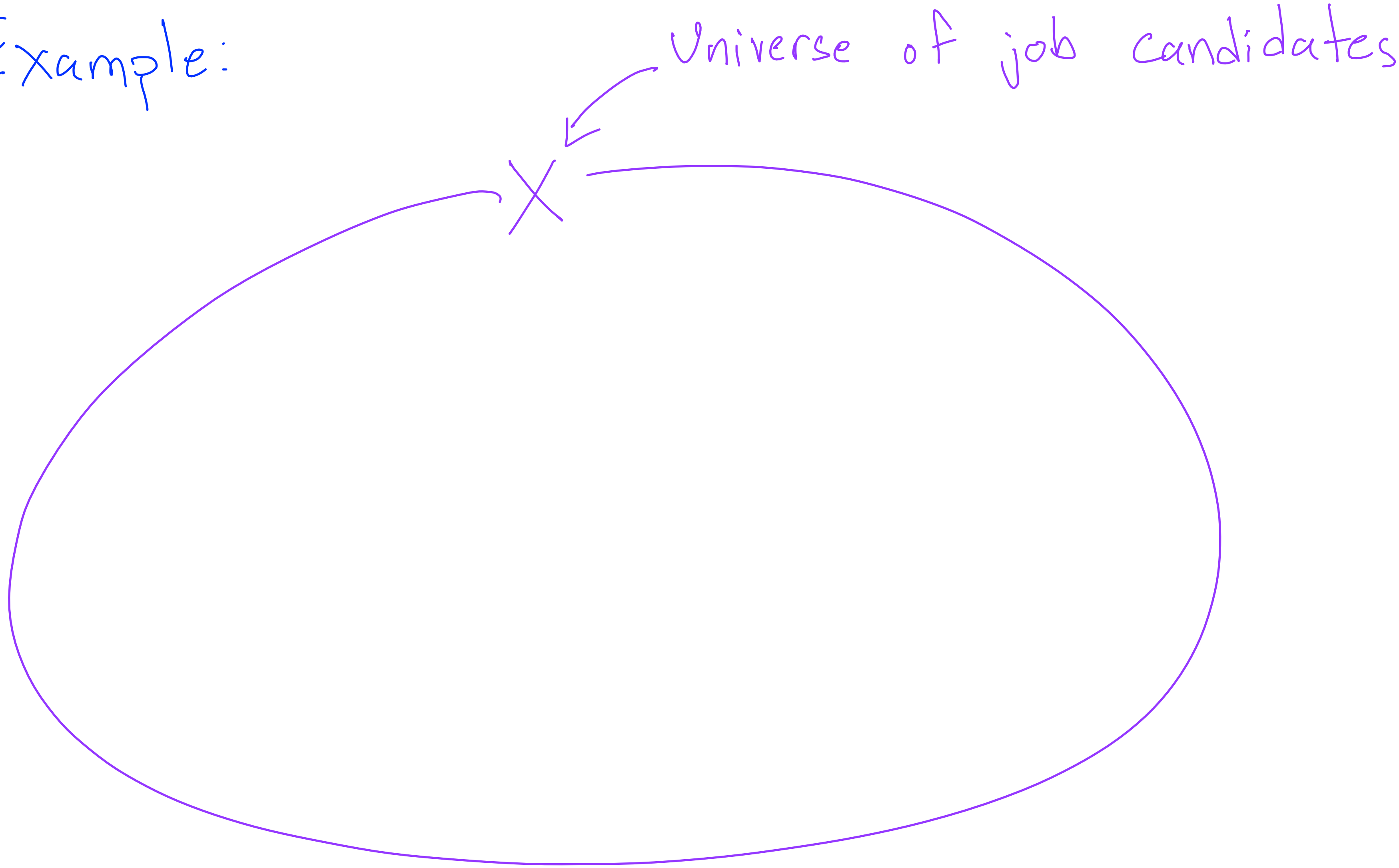
Example:

Organization has two divisions

$C_i \leftarrow$ Division i's hiring decisions

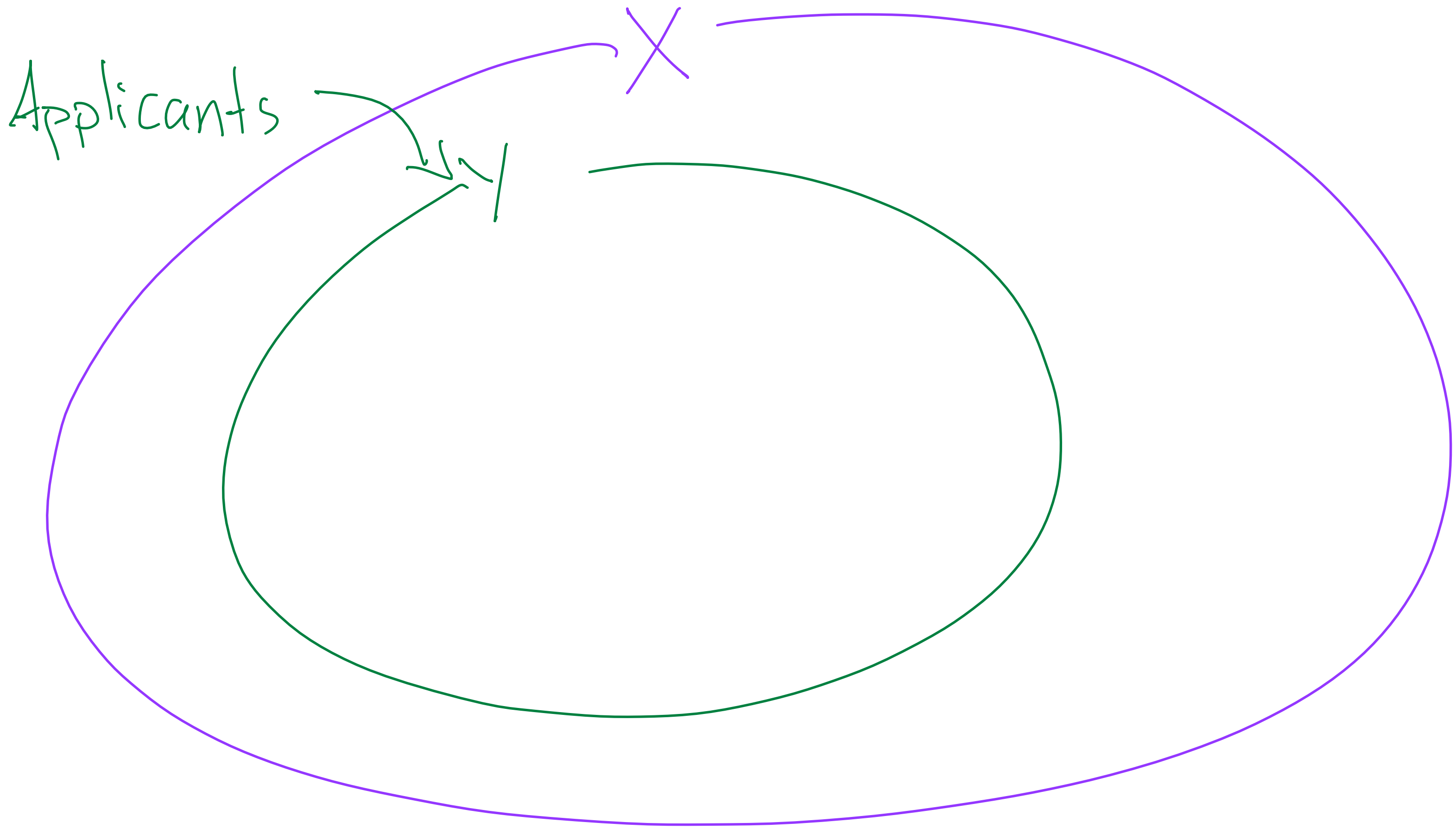
Sequential Composition

Example:



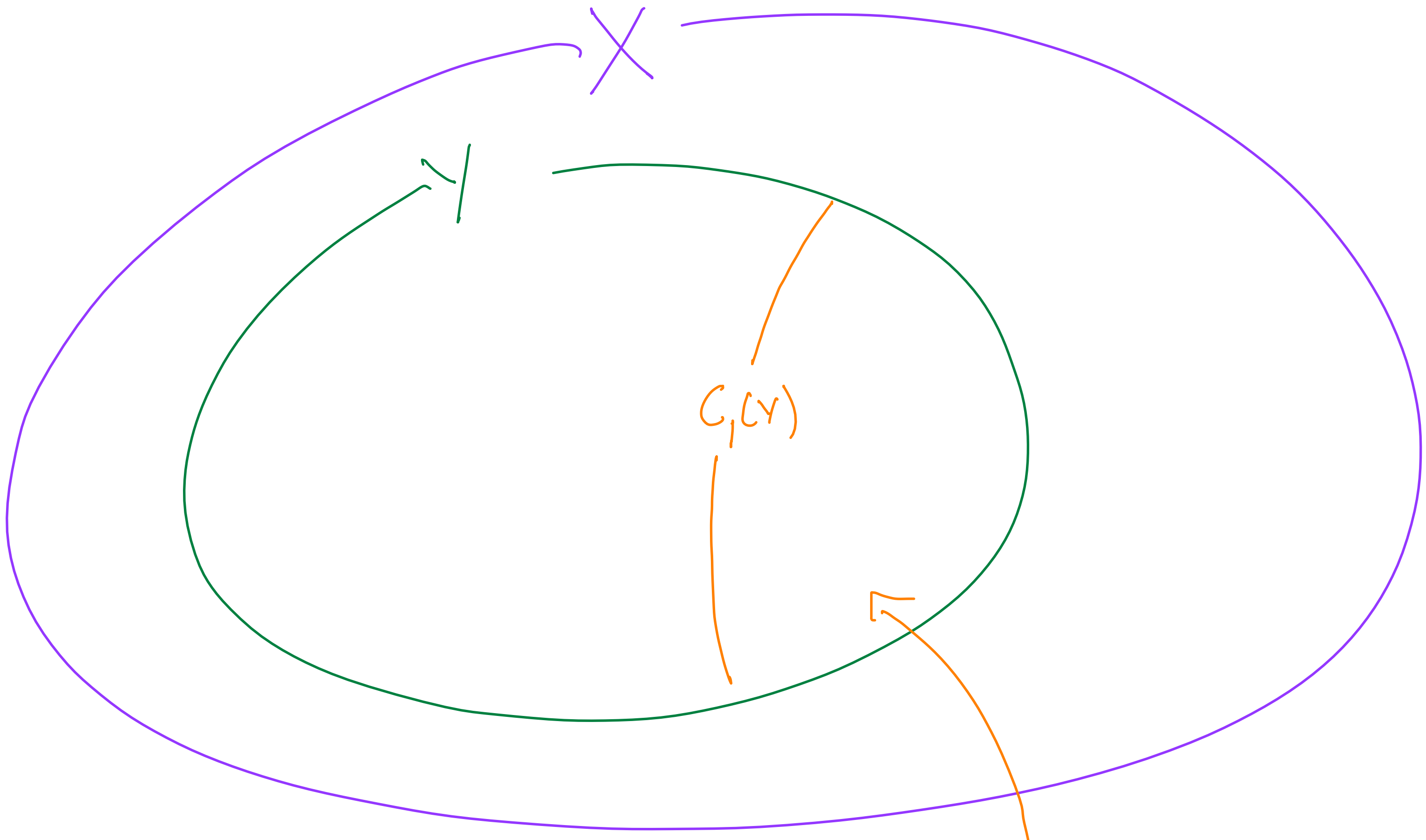
Sequential Composition

Example:



Sequential Composition

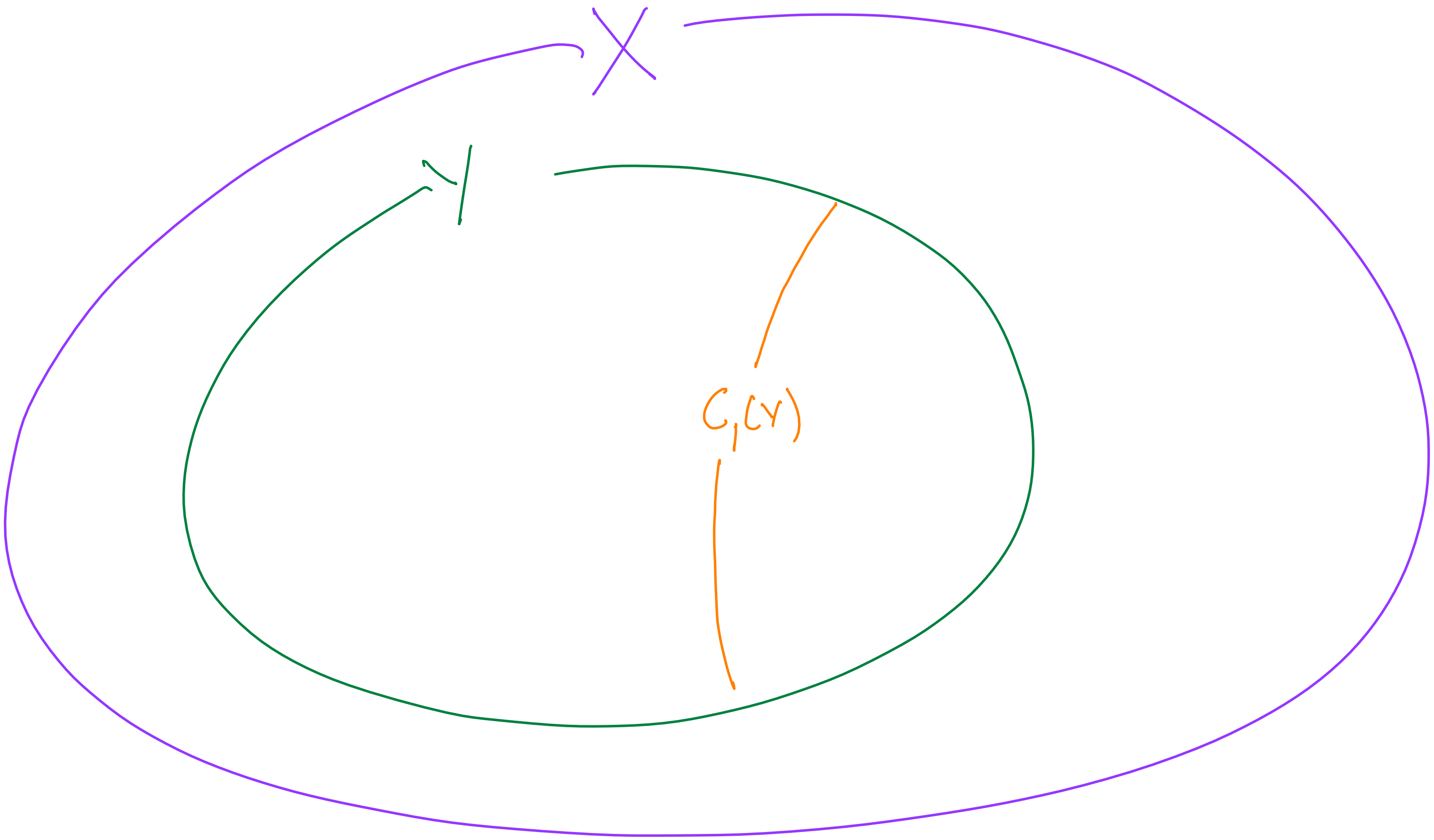
Example:



Division 1 chooses first and picks these applicants

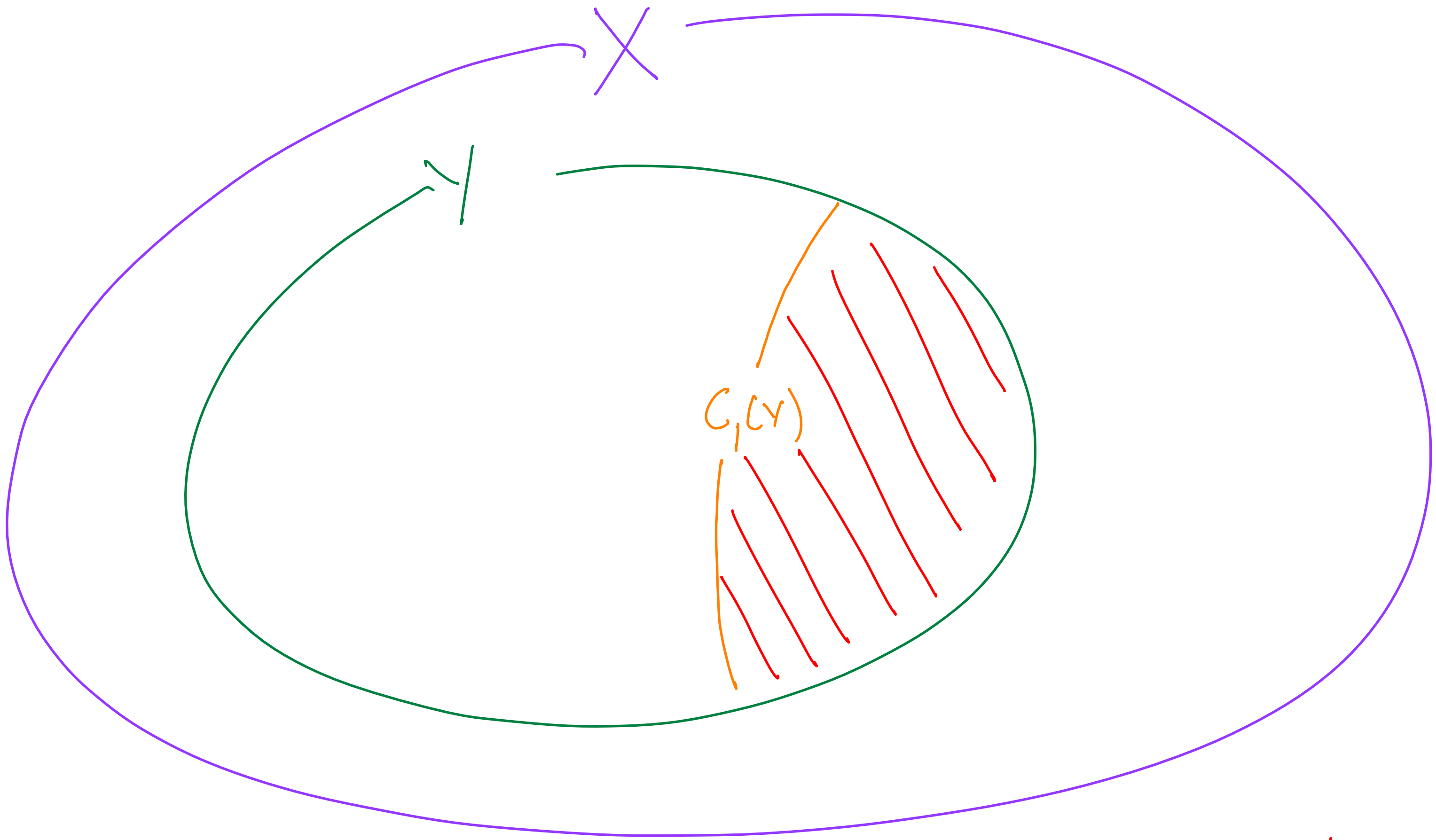
Sequential Composition

Example:



Sequential Composition

Example:

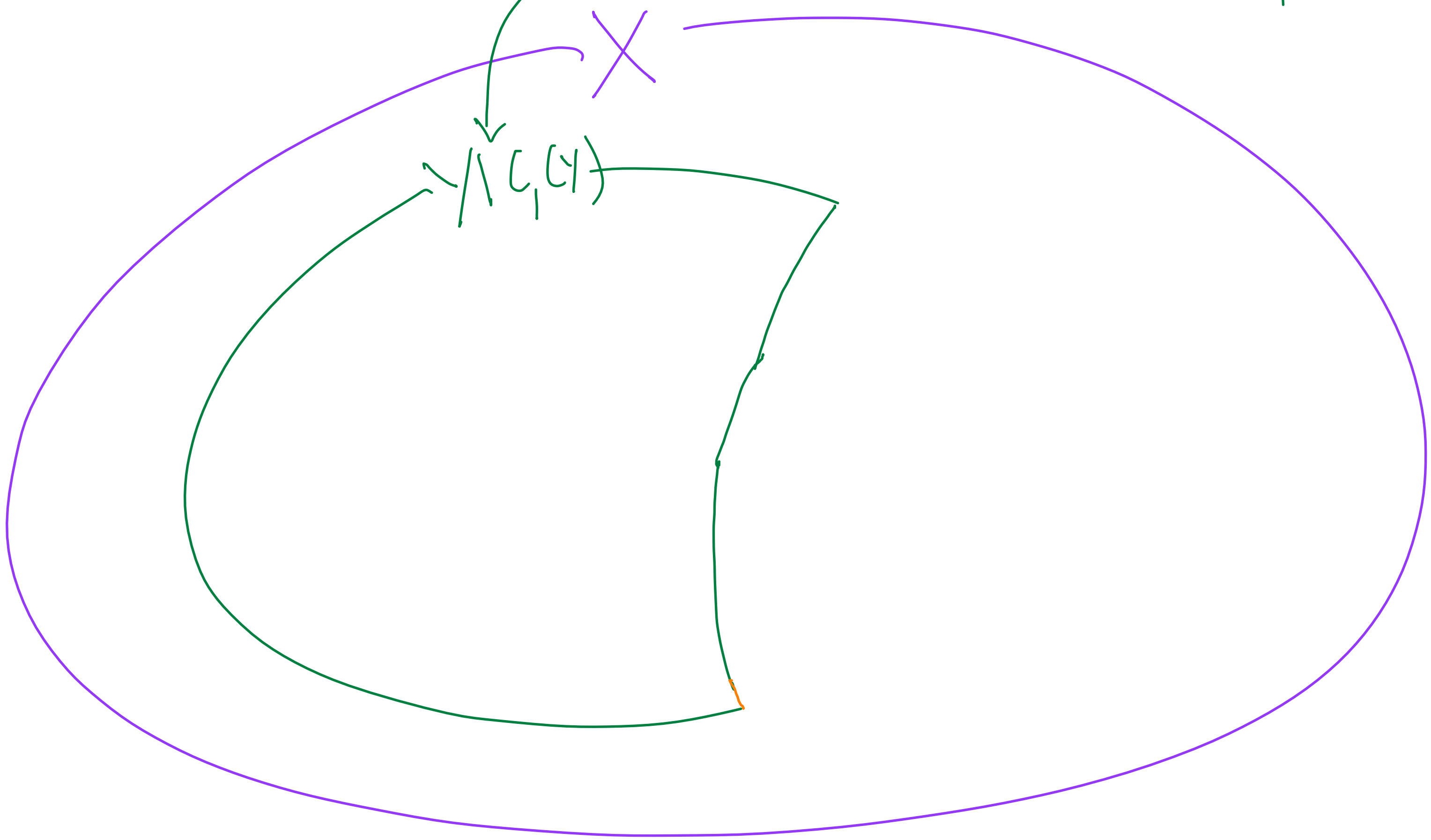


Division 2 can't pick the same ones

Sequential Composition

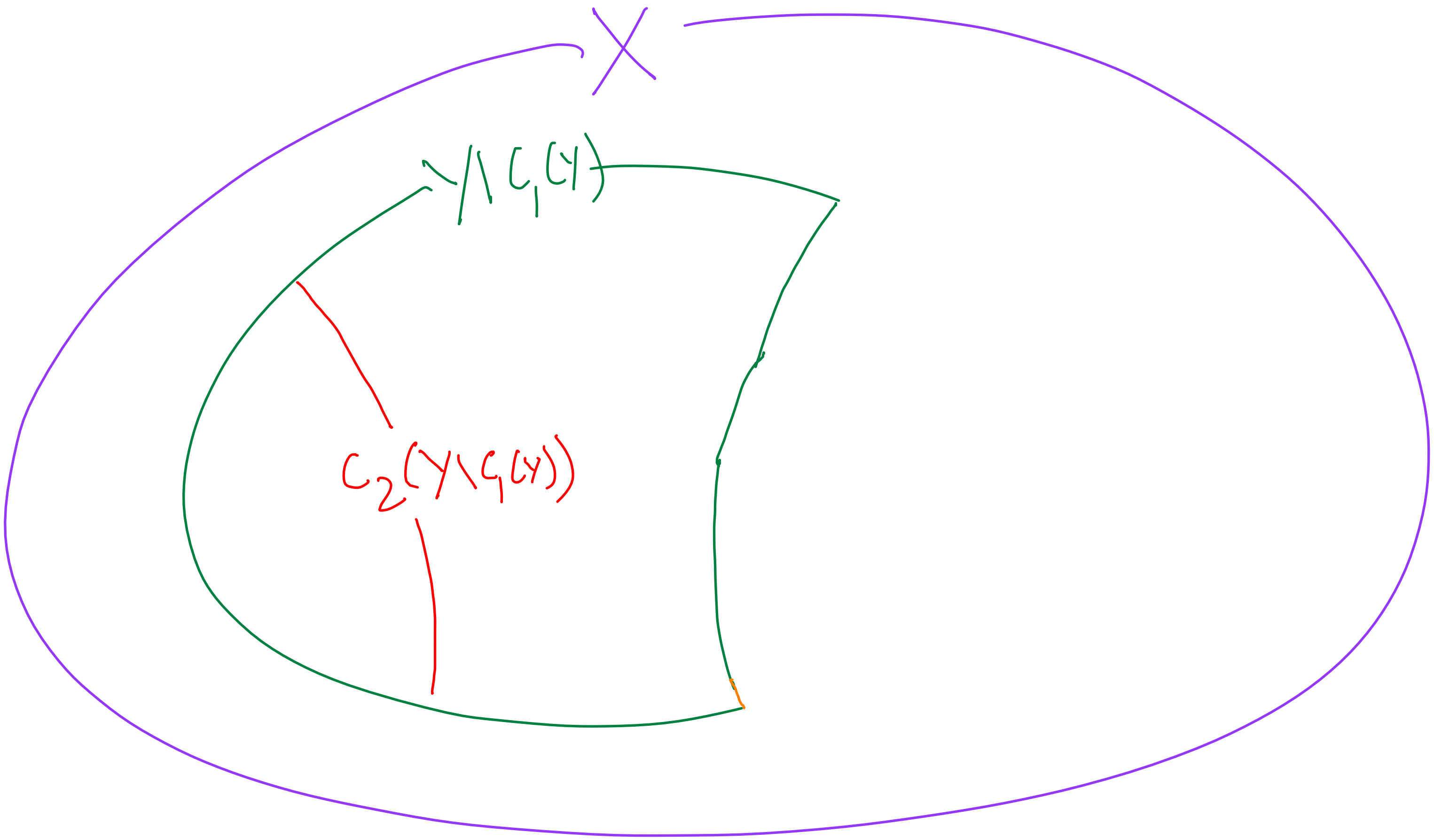
Example:

Applicants left for division 2 to choose from



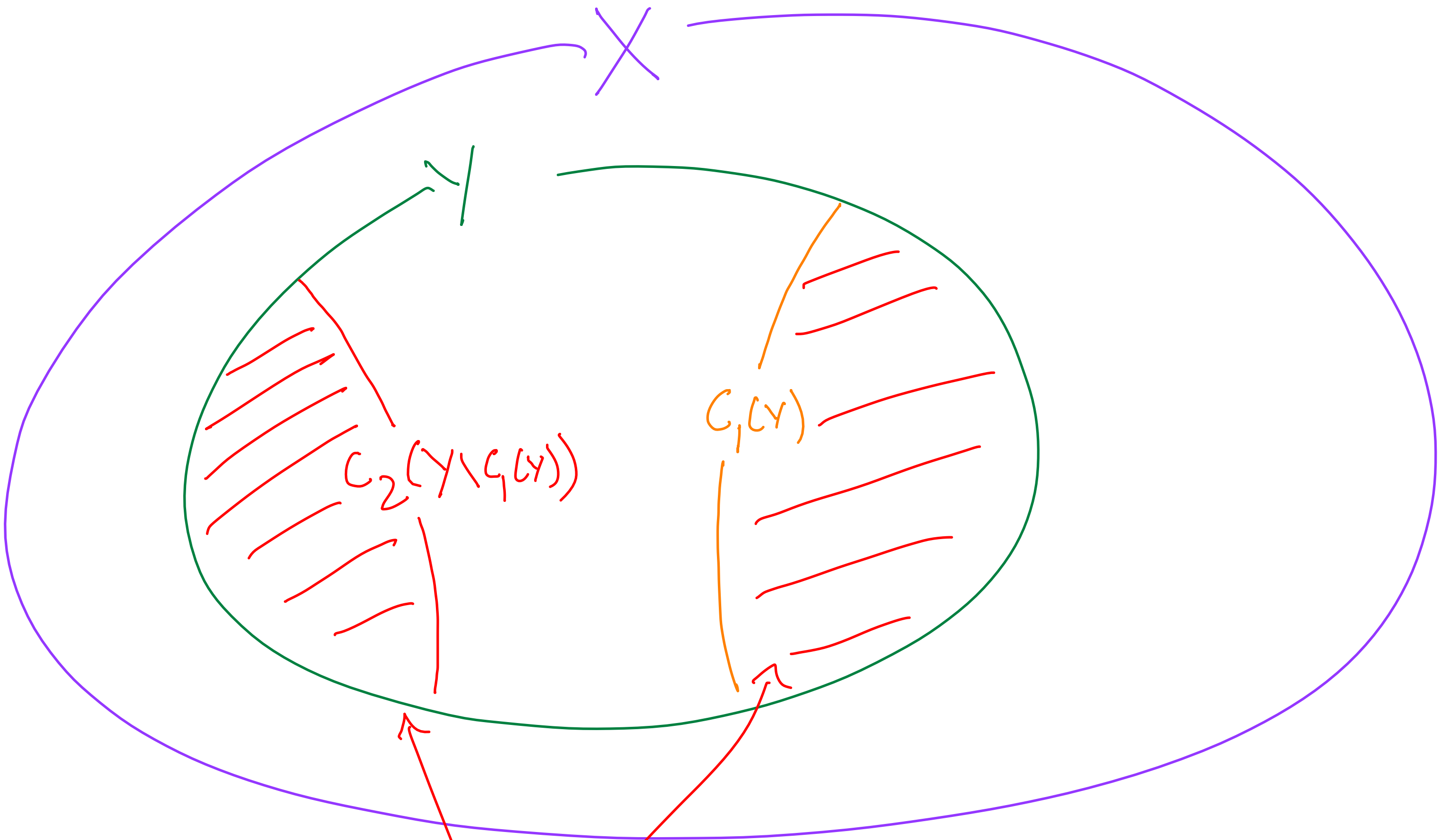
Sequential Composition

Example:



Sequential Composition

Example:



Aggregate choice of the organization

Sequential Composition

Example:

Candidates were "private" goods for the divisions

Sequential Composition

Example:

Candidates were "private" goods for the divisions

What if they are "public" goods instead?

Sequential Composition

Example:

Candidates were "private" goods for the divisions

What if they are "public" goods instead?

Generalized sequential composition handles that

Generalized Sequential Composition

$$\begin{tikzcd}
Y \arrow[d] \\
C_1 \arrow[d, "\text{picks}"'] \\
Z_1
\end{tikzcd}$$

Generalized Sequential Composition

Y

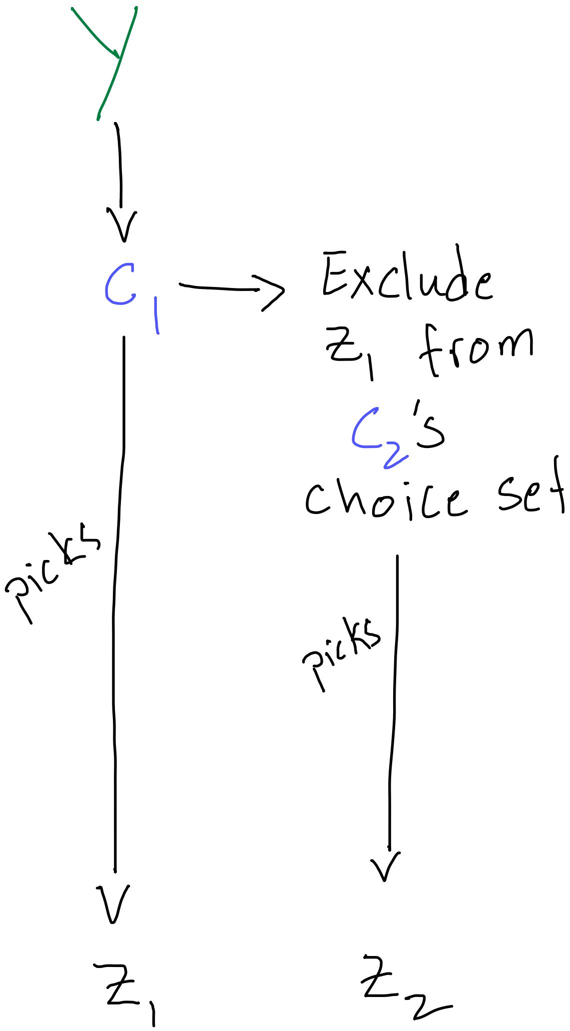
$\downarrow$

$C_1 \rightarrow$ Exclude z_1 from C_2's choice set

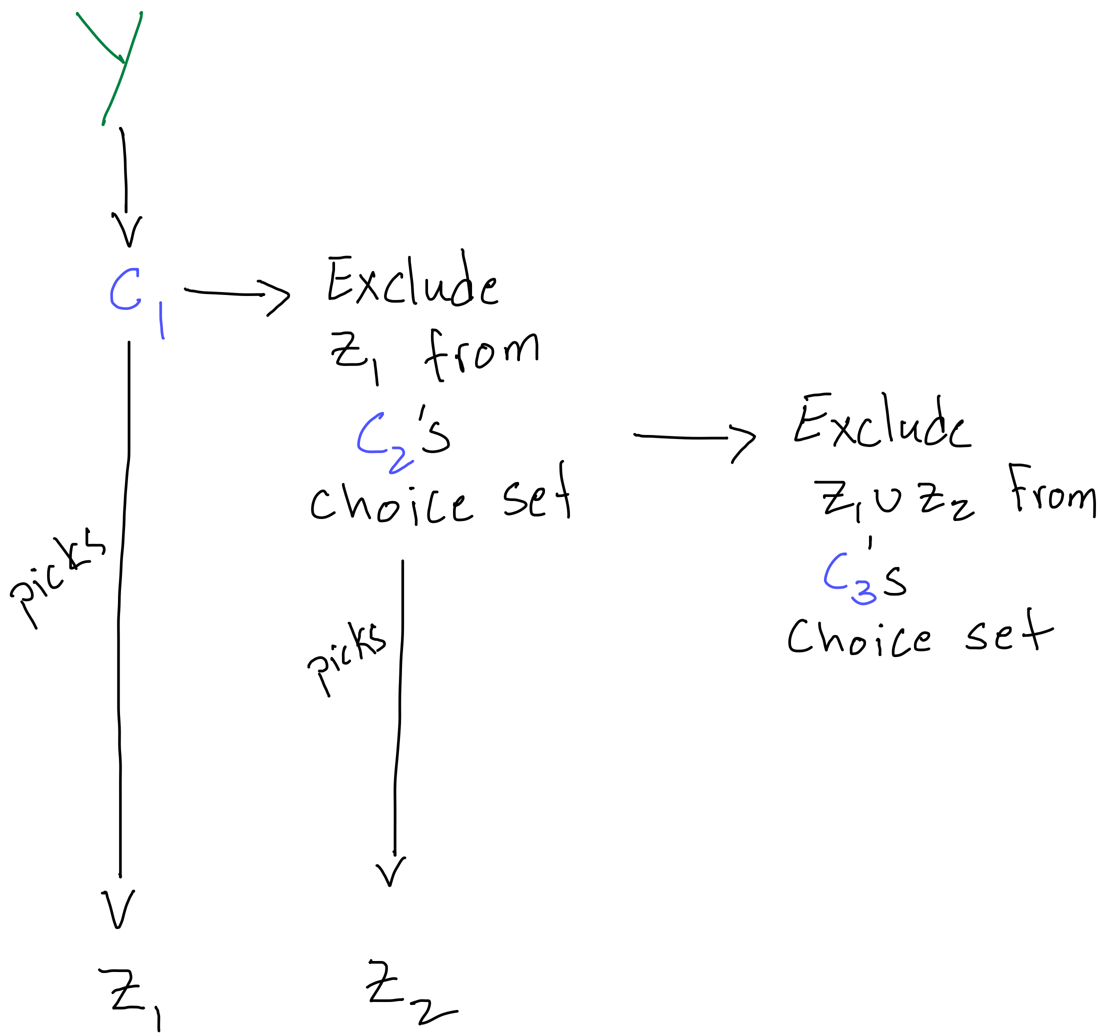
$\downarrow$ picks

z_1

Generalized Sequential Composition



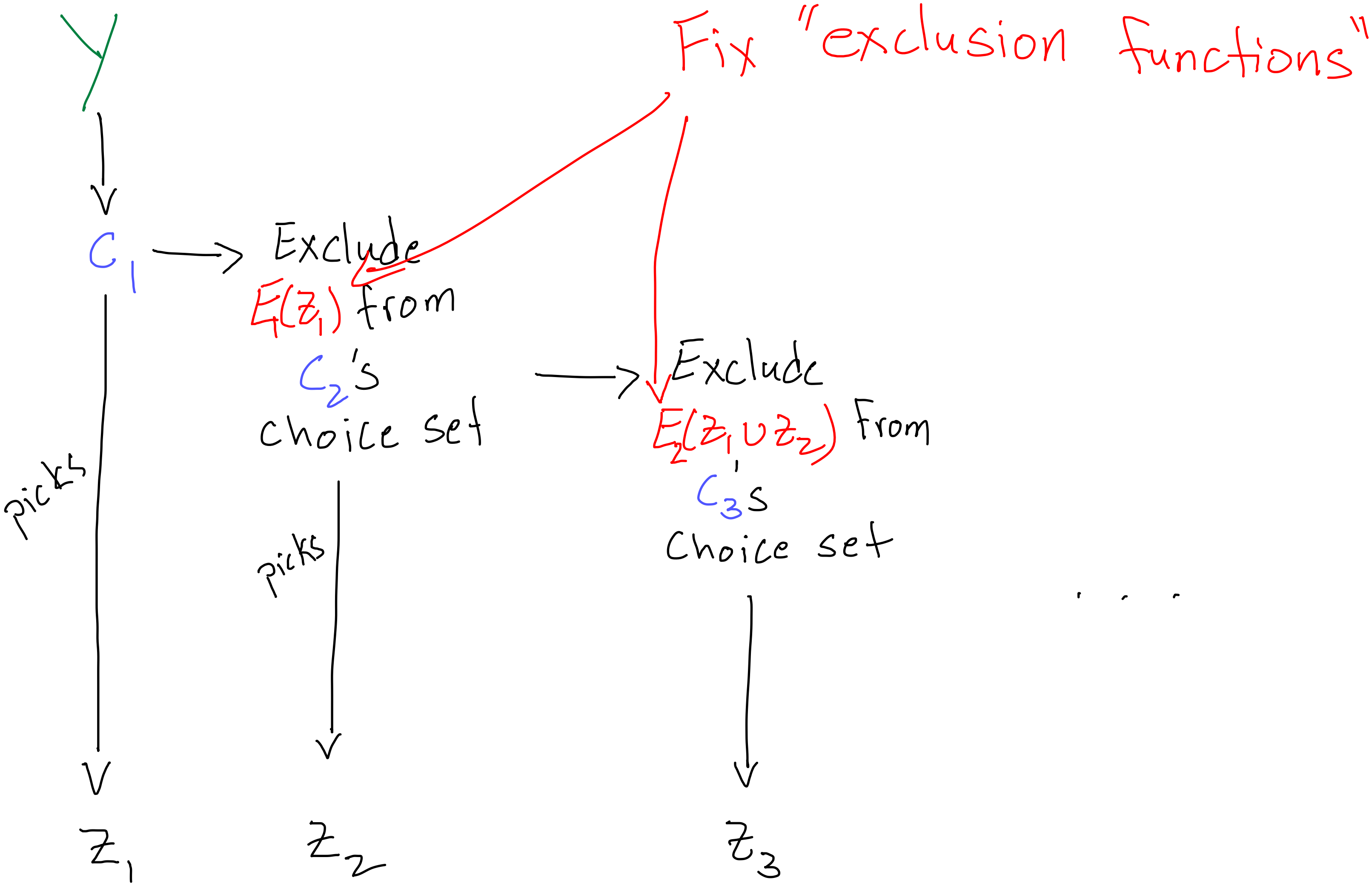
Generalized Sequential Composition



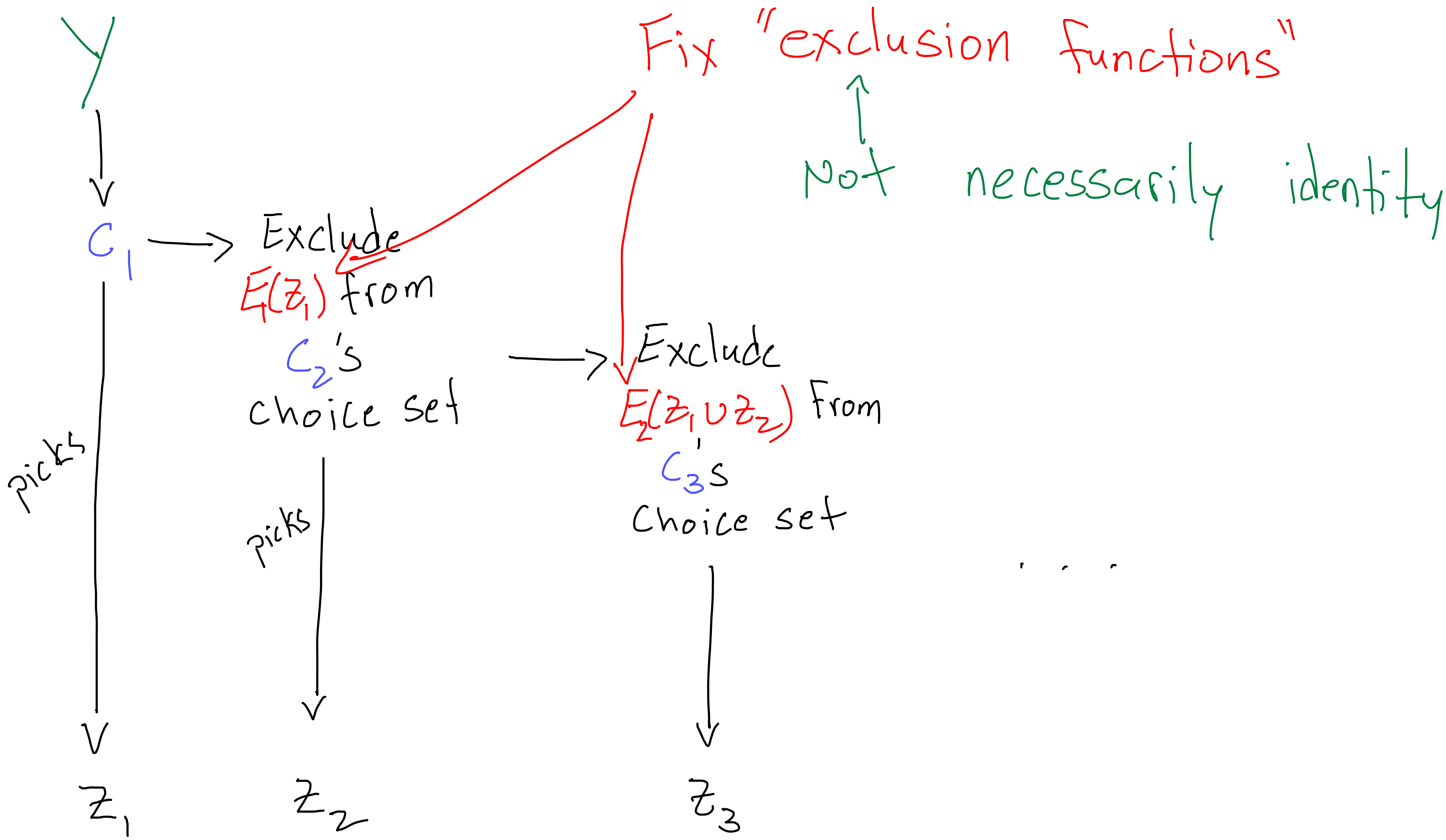
Generalized Sequential Composition

$$
\begin{tikzcd}
Y \arrow[d] & & \\
C_1 \arrow[r] \arrow[d, "\text{picks}"'] & \text{Exclude } z_1 \text{ from } C_2\text{'s choice set} \arrow[r] \arrow[d, "\text{picks}"'] & \text{Exclude } z_1 \cup z_2 \text{ from } C_3\text{'s choice set} \arrow[d] & \cdots \\
z_1 & z_2 & z_3 &
\end{tikzcd}
$$

Generalized Sequential Composition



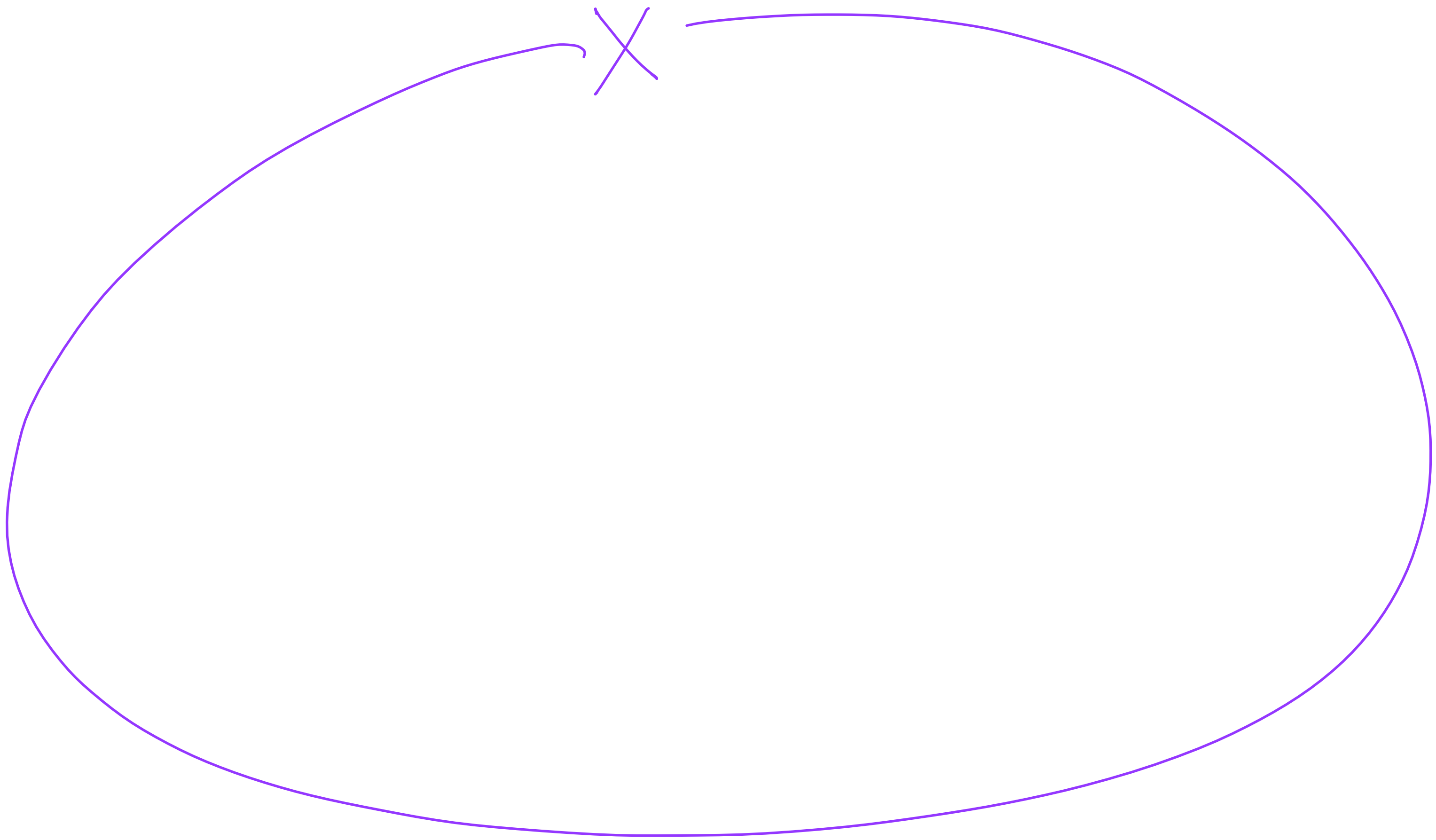
Generalized Sequential Composition



Generalized Sequential Composition

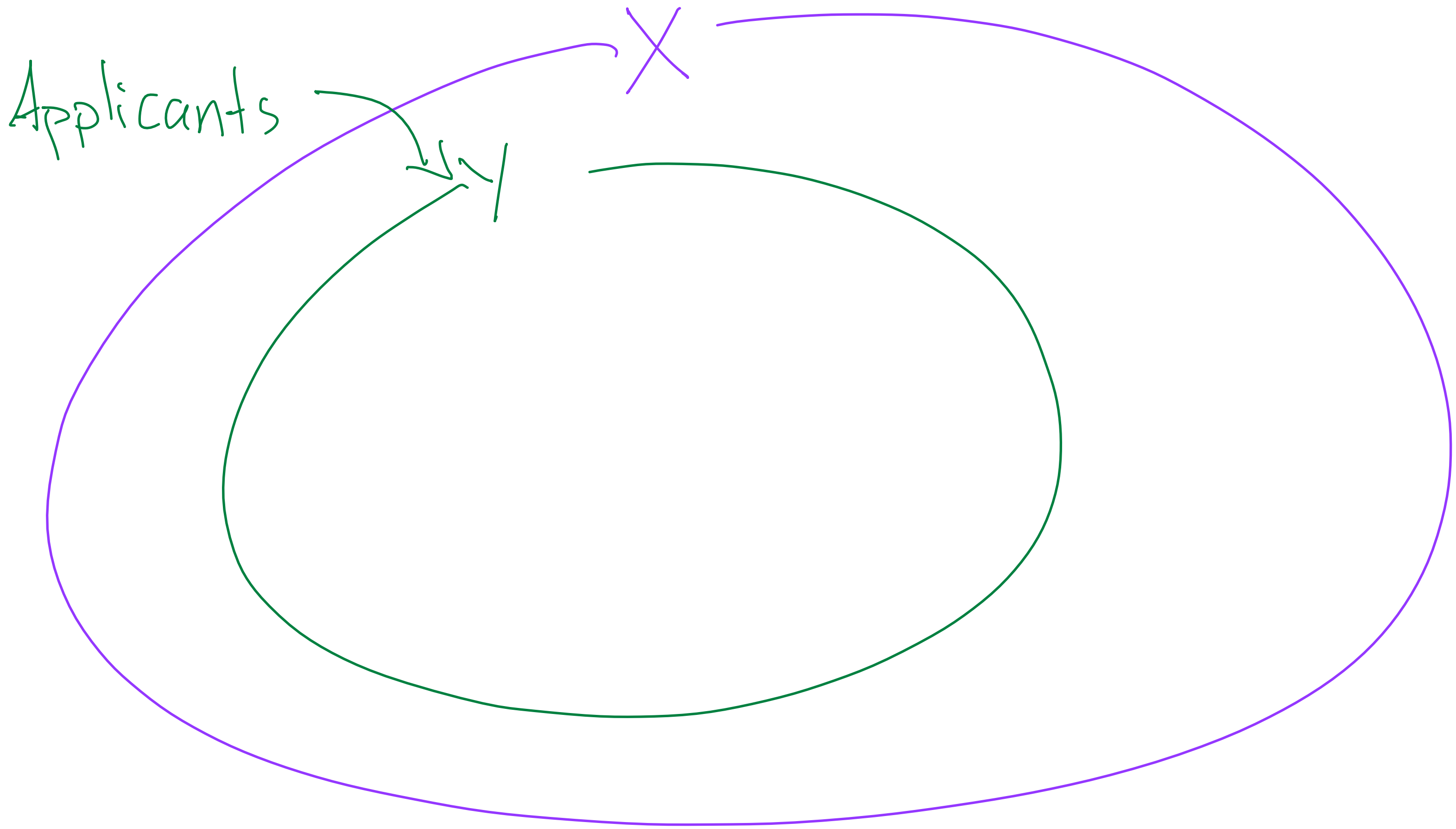
Example:

Universe of job candidates



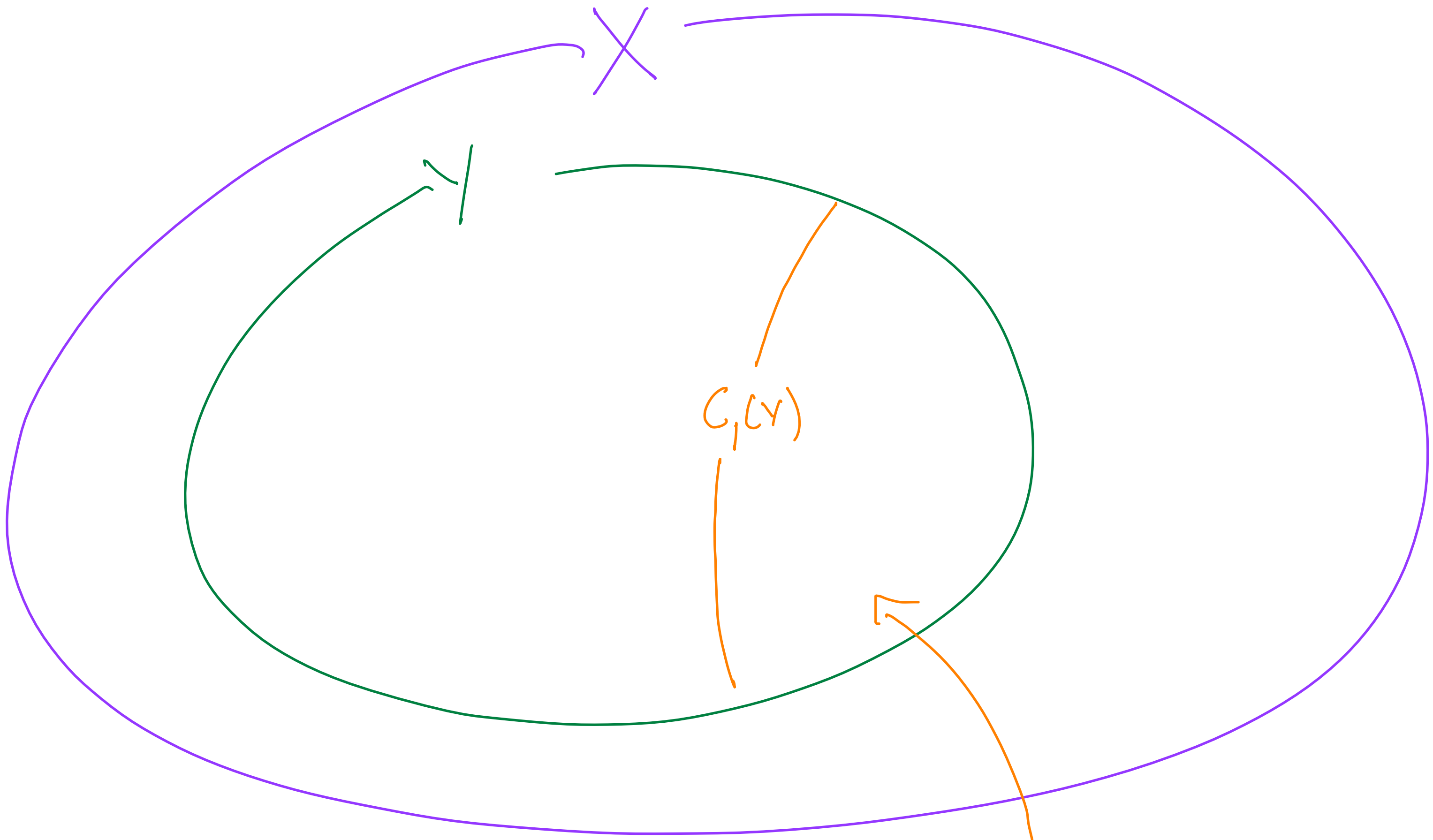
Generalized Sequential Composition

Example:



Generalized Sequential Composition

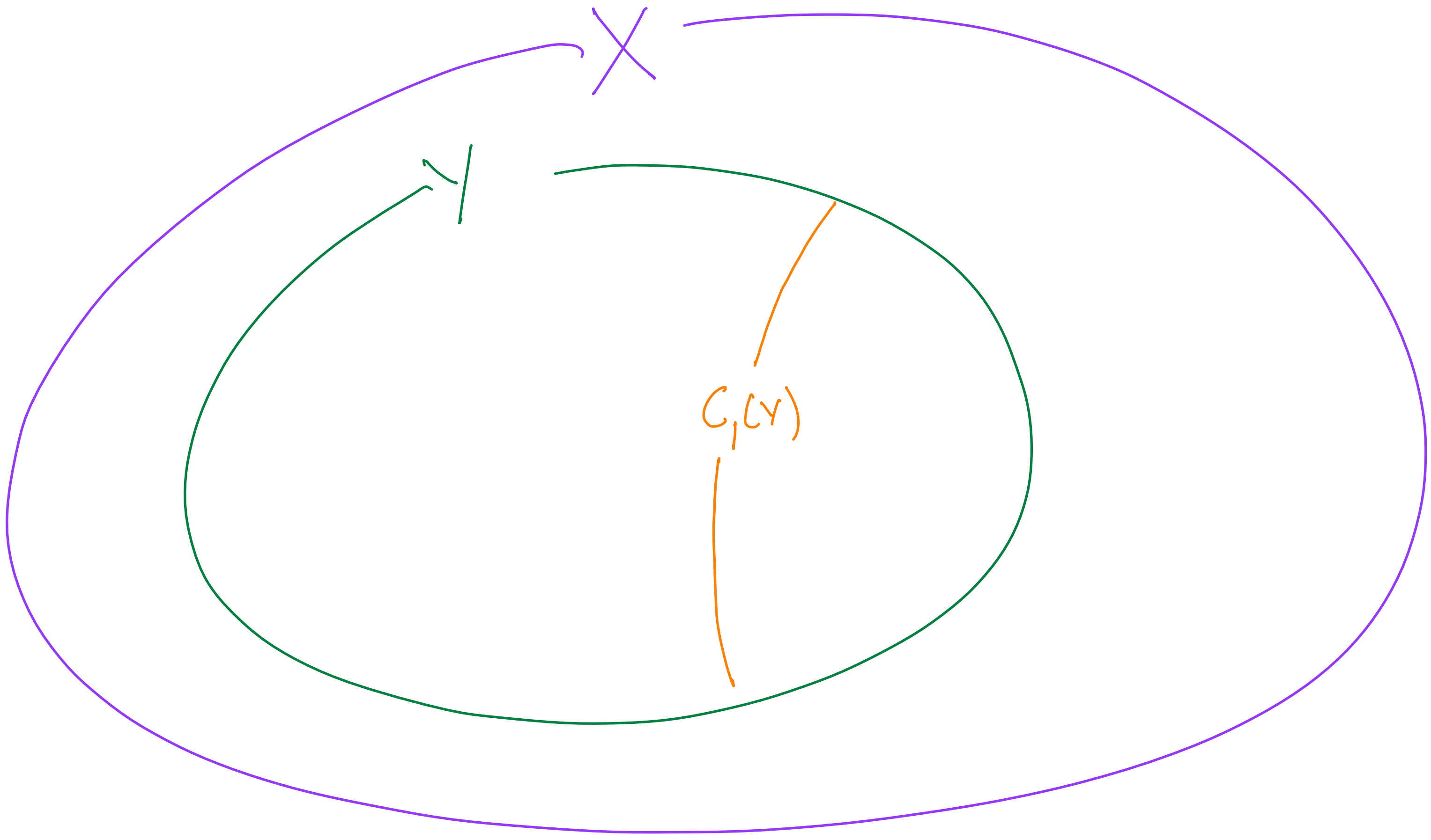
Example:



Division 1 chooses first and picks these applicants

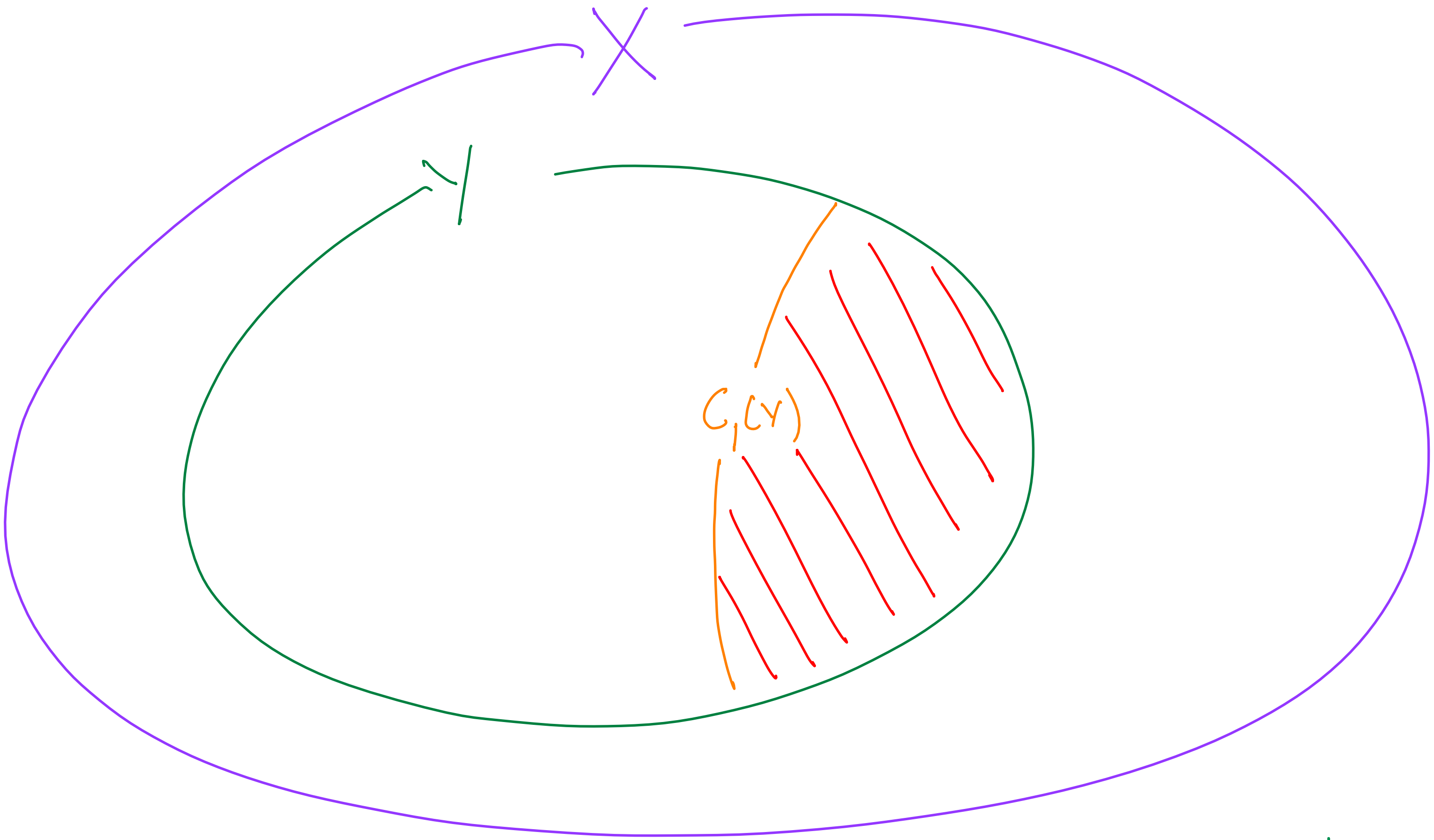
Generalized Sequential Composition

Example:



Generalized Sequential Composition

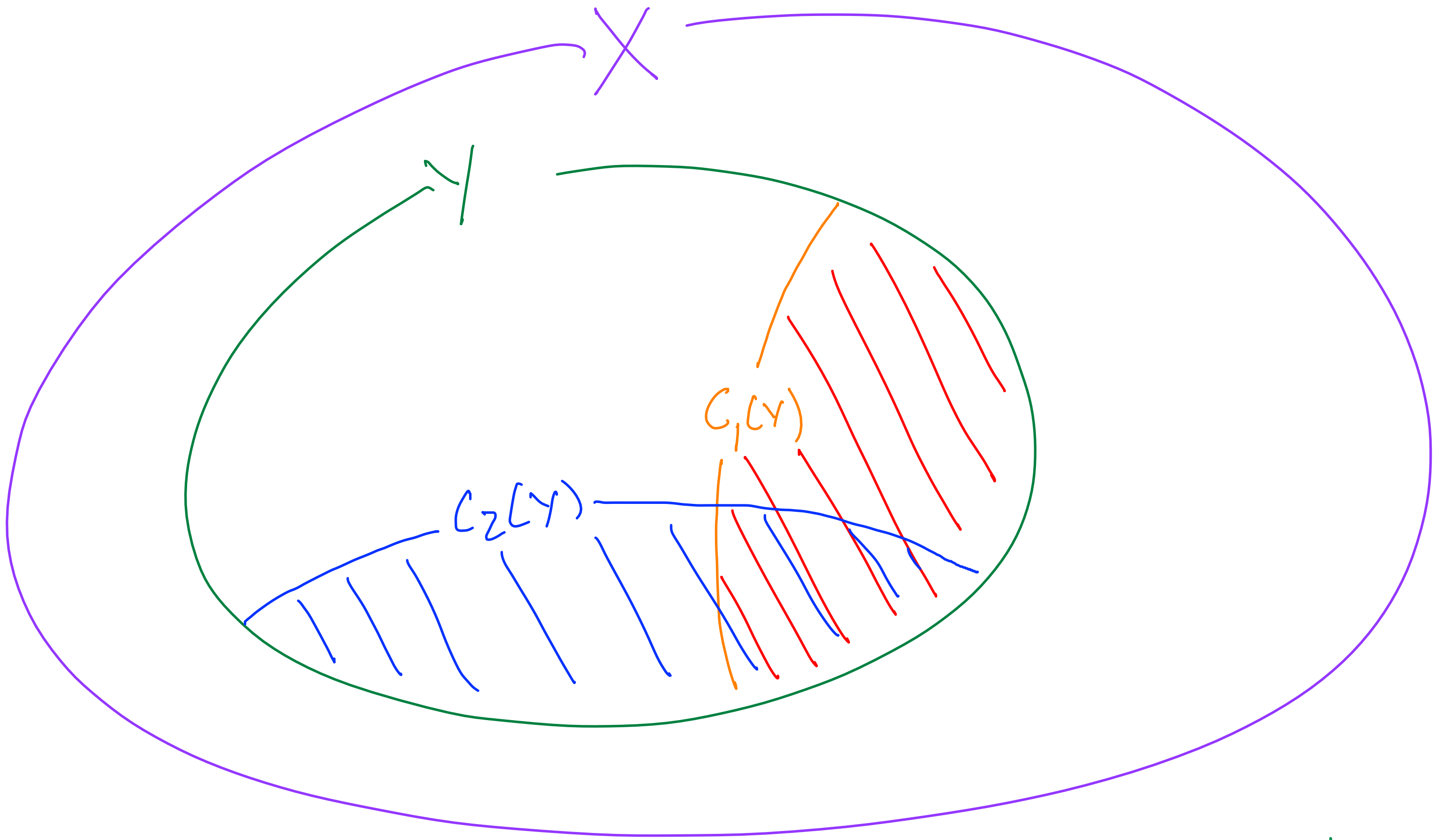
Example:



Division 2 <u>can</u> pick the same ones

Generalized Sequential Composition

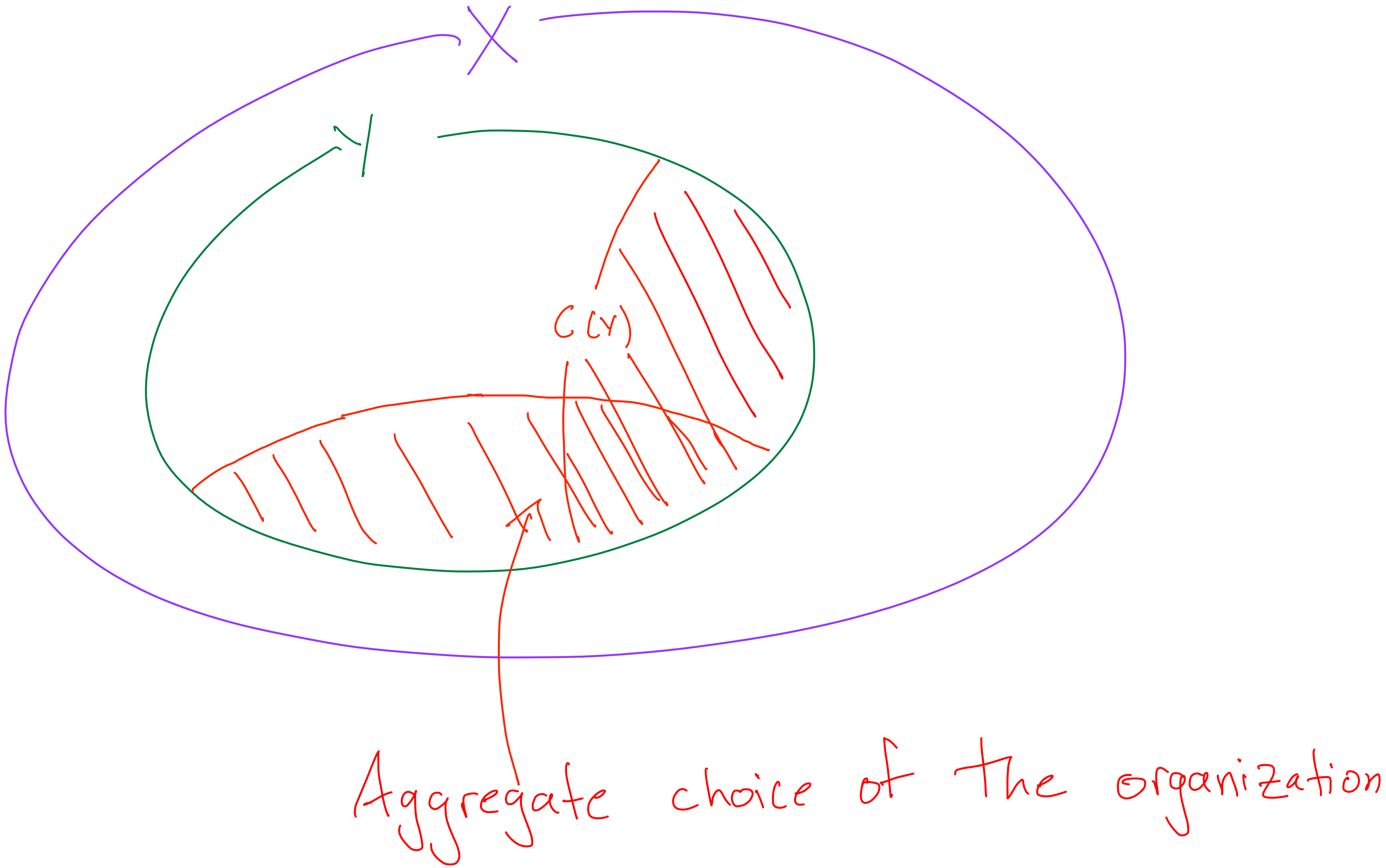
Example:



Division 2 <u>can</u> pick the same ones

Generalized Sequential Composition

Example:



Generalized Sequential Composition

More examples:

Generalized Sequential Composition

More examples:

- Matching with "contracts"

Generalized Sequential Composition

More examples:

- Matching with "contracts"

  Z already chosen

Generalized Sequential Composition

More examples:

- Matching with "contracts"

  Z already chosen $\longrightarrow$ Can't choose more contracts with doctors named in Z

Generalized Sequential Composition

More examples:

- Matching with "contracts"
- Affirmative action in school choice

Generalized Sequential Composition

More examples:

- Matching with "contracts"
- Affirmative action in school choice

  Z contains too many majority students

Generalized Sequential Composition

More examples:

- Matching with "contracts"
- Affirmative action in school choice

Z contains too many majority students $\longrightarrow$ Can only pick minorities

Generalized Sequential Composition

More examples:

- Matching with "contracts"
- Affirmative action in school choice
- Local public goods

Generalized Sequential Composition

More examples:

- Matching with "contracts"
- Affirmative action in school choice
- Local public goods

  z chosen by local authority

Generalized Sequential Composition

More examples:

- Matching with "contracts"
- Affirmative action in school choice
- Local public goods

Z chosen by local authority → residents can only pick from Z

Generalized Sequential Composition

More examples:

- Matching with "contracts"
- Affirmative action in school choice
- Local public goods

  Z chosen by local authority → residents can only pick from Z

  $(E(Z) = X \setminus Z)$

“Nice” Properties of Choice Functions

"Nice" Properties of Choice Functions

Path Independence (Plott 1973)

$$c(Y \cup Y') = c(c(Y) \cup c(Y'))$$

"Nice" Properties of Choice Functions

Path Independence (Plott 1973)

$$c(Y \cup Y') = c(c(Y) \cup c(Y'))$$

- Weaker than rationality

"Nice" Properties of Choice Functions

Path Independence (Plott 1973)

$$c(Y \cup Y') = c(c(Y) \cup c(Y'))$$

- Weaker than rationality

Equivalent to "pseudo-rationalizability" (Aizerman & Malishevski 1981)

"Nice" Properties of Choice Functions

Path Independence (Plott 1973)

$$c(Y \cup Y') = c(c(Y) \cup c(Y'))$$

- Weaker than rationality

Equivalent to "pseudo-rationalizability" (Aizerman & Malishevski 1981)

(Union of single-valued rationalizable choice functions (Moulin 1985))

"Nice" Properties of Choice Functions

Path Independence (Plott 1973)

$$c(Y \cup Y') = c(c(Y) \cup c(Y'))$$

- Weaker than rationality
- Normatively appealing

"Nice" Properties of Choice Functions

Path Independence (Plott 1973)

$$c(Y \cup Y') = c(c(Y) \cup c(Y'))$$

- Weaker than rationality
- Normatively appealing
  - Decisions can be made by considering smaller sets
  - Immunity to agenda manipulation
  - Computational "efficiency"

"Nice" Properties of Choice Functions

Path Independence (Plott 1973)

$$c(Y \cup Y') = c(c(Y) \cup c(Y'))$$

Standard Sequential Composition preserves PI

- Weaker than rationality
- Normatively appealing
  - Decisions can be made by considering smaller sets
  - Immunity to agenda manipulation
  - Computational "efficiency"

Do These Generalized Compositions Preserve Niceness?

Do These Generalized Compositions Preserve Niceness?

Not necessarily

Do These Generalized Compositions Preserve Niceness?

Not necessarily

Example in matching:

$E(Z) = \{$ Contracts that name a doctor who already has a contract in $Z \}$

Do These Generalized Compositions Preserve Niceness?

Not necessarily

Example in matching:

$E(Z) = \{$contracts that name a doctor who already has a contract in $Z\}$

Exclusion based on equivalence relation does not preserve path independence

(Hatfield & Milgrom 2005, Hatfield & Kojima 2010)

Do These Generalized Compositions Preserve Niceness?

Contribution of this paper:

Characterize Es that do preserve path independence

Definitions

X – universe of "items" (countably infinite)

Definitions

X – universe of "items"

$[X]$ – all subsets of X

Definitions

X – universe of "items"

$[X]^*$ – all finite subsets of X (menus)

Definitions

X - universe of "items"

$[X]^*$ - all finite subsets of X (menus)

$S : [X]^* \longrightarrow [X]$ - set function

Definitions

X – universe of "items"

$[X]^*$ – all finite subsets of X (menus)

$S : [X]^* \longrightarrow [X]$ – set function

$\mathcal{S}$ – all set functions

Definitions

X – universe of "items"

$[X]^*$ – all finite subsets of X (menus)

$S : [X]^* \longrightarrow [X]$ – set function

$\mathcal{S}$ – all set functions

$D \in \mathcal{S}$ such that $D(Y) \supseteq Y \quad \forall Y \in [X]^*$

– Dilation

Definitions

X – universe of "items"

$[X]^*$ – all finite subsets of X (menus)

$S : [X]^* \longrightarrow [X]$ – set function

$\mathcal{S}$ – all set functions

$D \in \mathcal{S}$ such that $D(Y) \supseteq Y \quad \forall Y \in [X]^*$

– Dilation

$\mathcal{D}$ – all dilations

Definitions

X – universe of "items"

$[X]^*$ – all finite subsets of X (menus)

$S : [X]^* \longrightarrow [X]$ – set function

$\mathcal{S}$ – all set functions

$C \in \mathcal{S}$ such that $C(Y) \subseteq Y \;\; \forall Y \in [X]^*$

– Contraction

$\mathcal{C}$ – all contractions

Definitions

X - universe of "items"

$[X]^*$ - all finite subsets of X (menus)

$S : [X]^* \longrightarrow [X]$ - set function

$\mathcal{S}$ - all set functions

$C \in \mathcal{S}$ such that $C(Y) \subseteq Y \quad \forall Y \in [X]^*$

- Choice function

$\mathcal{C}$ - all choice functions

Definitions

$c_1, c_2 \in \mathcal{C}$ ← choice functions

$E \in \mathcal{E}$ ← exclusion function

Sequential composition of c_1 and c_2 subject to E:

$$\Sigma_E(c_1, c_2) \in \mathcal{C}$$

Definitions

$c_1, c_2 \in \mathcal{C}$ ← choice functions

$E \in \mathcal{E}$ ← exclusion function

Sequential composition of c_1 and c_2 subject to E:

$$\forall Y \in [X]^*$$

$$\sum_E (c_1, c_2)$$

Definitions

$c_1, c_2 \in \mathcal{C}$ ← choice functions

$E \in \mathcal{E}$ ← exclusion function

Sequential composition of c_1 and c_2 subject to E:

$\forall Y \in [X]^*$

$$\sum_E (c_1, c_2)(Y)$$

Definitions

$c_1, c_2 \in \mathcal{C}$ ← choice functions

$E \in \mathcal{E}$ ← exclusion function

Sequential composition of c_1 and c_2 subject to E:

$\forall Y \in [X]^*$

$\sum_E (c_1, c_2)(Y) =$

Definitions

$c_1, c_2 \in \mathcal{C}$ ← choice functions

$E \in \mathcal{E}$ ← exclusion function

Sequential composition of c_1 and c_2 subject to E:

$\forall Y \in [X]^*$

$$\sum_E (c_1, c_2)(Y) = c_1(Y)$$

Definitions

$c_1, c_2 \in \mathcal{C} \leftarrow$ choice functions

$E \in \mathcal{E} \leftarrow$ exclusion function

Sequential composition of c_1 and c_2 subject to E:

$\forall Y \in [X]^*$

$$\sum_E (c_1, c_2)(Y) = c_1(Y) \cup c_2(\quad)$$

Definitions

$c_1, c_2 \in \mathcal{C}$ ← choice functions

$E \in \mathcal{E}$ ← exclusion function

Sequential composition of c_1 and c_2 subject to E:

$\forall Y \in [X]^*$

$$\Sigma_E(c_1, c_2)(Y) = c_1(Y) \cup c_2(Y \setminus E(c_1(Y)))$$

Definitions

$c_1, c_2 \in \mathcal{C}$ ← choice functions

$E \in \mathcal{E}$ ← exclusion function

Sequential composition of c_1 and c_2 subject to E:

$$\forall Y \in [X]^*$$

$$\Sigma_E(c_1, c_2)(Y) = c_1(Y) \cup c_2(Y \setminus E(c_1(Y)))$$

Definitions

$\mathcal{C}^{PI}$ - path independent choice functions

Definitions

$\mathcal{C}^{PI}$ - path independent choice functions

Σ_E preserves PI over $\mathcal{C}' \subseteq \mathcal{C}^{PI}$ if

Definitions

$\mathcal{C}^{PI}$ - path independent choice functions

Σ_E preserves PI over $\mathcal{C}' \subseteq \mathcal{C}^{PI}$ if

$\forall c_1, c_2 \in \mathcal{C}'$

$\Sigma_E(c_1, c_2)$ is path independent

Definitions

$\mathcal{C}^{res} \subset \mathcal{C}^{PI}$ – "Responsive" choice functions (Roth & Sotomayor 1990)

Definitions

$\mathcal{C}^{res} \subset \mathcal{C}^{PI}$ – "Responsive" choice functions (Roth & Sotomayor 1990)

$\forall\, c \in \mathcal{C}^{res}$ $\exists$ $\succ$, q such that

complete, antisymmetric, transitive binary relation over $X \cup \{\emptyset\}$

Definitions

$\mathcal{C}^{res} \subset \mathcal{C}^{PI}$ – "Responsive" choice functions
(Roth & Sotomayor 1990)

$\forall c \in \mathcal{C}^{res} \quad \exists \succ, q$ such that

integer

Definitions

$\mathcal{C}^{res} \subset \mathcal{C}^{PI}$ – "Responsive" choice functions (Roth & Sotomayor 1990)

$\forall c \in \mathcal{C}^{res}$ $\exists \succ, q$ such that

$\forall Y \in [X]^*$, $c(Y)$ is the q $\succ$-best items in Y (that beat $\emptyset$)

Definitions

$\mathcal{C}^{res} \subset \mathcal{C}^{PI}$ – "Responsive" choice functions (Roth & Sotomayor 1990)

$\forall c \in \mathcal{C}^{res} \quad \exists \succ, q \quad$ such that

$\forall Y \in [X]^*, \; c(Y)$ is the q $\succ$-best items in Y

Start with preserving path independence over $\mathcal{C}^{res}$.

Definitions

$\mathcal{C}^{res} \subset \mathcal{C}^{PI}$ – "Responsive" choice functions (Roth & Sotomayor 1990)

$\forall c \in \mathcal{C}^{res} \quad \exists \succ, q$ such that

$\forall Y \in [X]^*, \; c(Y)$ is the q $\succ$-best items in Y

Start with preserving path independence over $\mathcal{C}^{res}$.

Necessary conditions on $\mathcal{C}^{res}$ are necessary conditions for any $\mathcal{C}' \supseteq \mathcal{C}^{res}$

Pure Expansion

Pure Expansion

E is a dilation

Monotonicity:

Pure Expansion

E is a dilation

Monotonicity:

$\forall Z, Z' \in [X]^*$

$$Z \leq Z' \Rightarrow E(Z) \subseteq E(Z')$$

Pure Expansion

All-or-nothing:

Pure Expansion

All-or-nothing:

$\forall z \in [X]^*$

$E(z) \in \{ X$

"all"

Pure Expansion

All-or-nothing:

$$\forall z \in [X]^* \quad E(z) \in \{X, \quad E(\{\})\}$$

"nothing"

Pure Expansion

All-or-nothing:

$$\forall z \in [X]^*$$

$$E(z) \in \{X,\ z \cup E(\{\})\}$$

Pure Expansion

Cardinal:

$$\forall Z, Z' \in [X]^*$$

$$\left.\begin{array}{l} Z \cup E(\{\}) \neq X \\ |Z| = |Z'| \\ E(Z) = X \end{array}\right\} \implies E(Z') = X$$

t-Exclusion

t-Exclusion

$t \in \mathbb{N} \cup \{0, \infty\} \leftarrow$ "threshold"

t-Exclusion

$t \in \mathbb{N} \cup \{0, \infty\}$

$\forall z \in [X]^*$

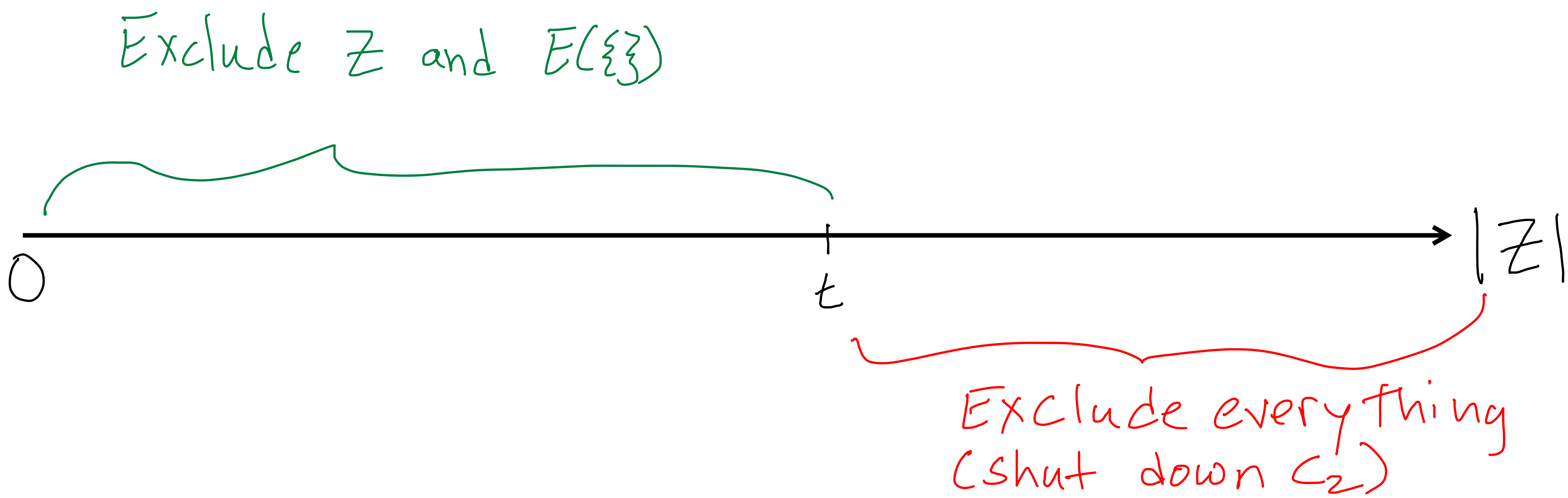
$$E(z) = \begin{cases} z \cup E(\{\}) & \text{if } |z| < t \\ X & \text{otherwise} \end{cases}$$

t-Exclusion

$t \in \mathbb{N} \cup \{0, \infty\}$

$\forall Z \in [X]^*$

$$E(Z) = \begin{cases} Z \cup E(\{\}) & \text{if } |Z| < t \\ X & \text{otherwise} \end{cases}$$



t-Exclusion

$t \in \mathbb{N} \cup \{0, \infty\}$

$\forall\, Z \in [X]^*$

$$E(Z) = \begin{cases} Z \cup E(\{\}) & \text{if } |Z| < t \\ X & \text{otherwise} \end{cases}$$

$$\left.\begin{array}{c} \text{Monotonicity} \\ + \\ \text{All-or-nothingness} \\ + \\ \text{Cardinality} \end{array}\right\} \iff \text{t-Exclusion}$$

t-Exclusion

$t \in \mathbb{N} \cup \{0, \infty\}$

$\forall Z \in [X]^*$

$$E(Z) = \begin{cases} Z \cup E(\{\}) & \text{if } |Z| < t \\ X & \text{otherwise} \end{cases}$$

Monotonicity

$+$

All-or-nothingness

$+$

Cardinality

$\Bigg\} \Longleftrightarrow$ t-Exclusion

Necessary to preserve PI over $\mathcal{E}^{res}$

t-Exclusion

Also sufficient

t-Exclusion

Also sufficient

Proposition: $\forall E \in \mathcal{D}$,

$$\Sigma_E \text{ preserves PI over } \mathcal{C}^{res}$$

$$\Updownarrow$$

E is a t-Exclusion

Pure Reuse

Pure Reuse

E is a contraction

Pure Reuse

E is a contraction

$I \setminus E$ also a contraction

Identity $(I(z) = z)$

Pure Reuse

E is a contraction

I \ E also a contraction

Conditions on I \ E rather than E

Pure Reuse

$$T^n = \{ x \in X \mid \exists Z \ni x \text{ s.t. } |Z| = n \text{ and } x \notin I \setminus E(Z) \}$$

All items that can be reused when chosen as a part of <u>some</u> set of n items

Pure Reuse

$$T^n = \{ x \in X \mid \exists Z \ni x \text{ s.t. } |Z| = n \text{ and } x \notin I \setminus E(Z) \}$$

$$T^0 = \{\}$$

Pure Reuse

$$T^n = \{ x \in X \mid \exists z \ni x \text{ s.t. } |z| = n \text{ and } x \notin I \setminus E(z) \}$$

$$T^0 = \{\}$$

$$T^{n-1} \subseteq T^n$$

Pure Reuse

$$T^n = \{ x \in X \mid \exists z \ni x \text{ s.t. } |z| = n \text{ and } x \notin I \setminus E(z) \}$$

$$T^0 = \{\}$$

$$T^{n-1} \subseteq T^n$$

Denote such a sequence by τ

Pure Reuse

$$T^n = \{ x \in X \mid \exists z \ni x \text{ s.t. } |z| = n \text{ and } x \notin IVE(z) \}$$

$$T^0 = \{\}$$

$$T^{n-1} \subseteq T^n$$

$$IVE(z) = z \cap T^{|z|}$$

Pure Reuse

$$T^n = \{ x \in X \mid \exists z \ni x \text{ s.t. } |z| = n \text{ and } x \notin I \setminus E(z) \}$$

$$T^0 = \{\}$$

$$T^{n-1} \subseteq T^n$$

Call such
a contraction
a $\hat{T}$-contraction

$$I \setminus E(z) = z \cap T^{|z|}$$

Pure Reuse

Proposition: $\forall E \in \mathcal{C}$,

$$\Sigma_E \text{ preserves PI over } \mathcal{C}^{res}$$

$$\Updownarrow$$

$I \backslash E$ is a Υ-contraction

Exclusion and Reuse

Exclusion and Reuse

$E \in \mathcal{J}$ – any set function

Exclusion and Reuse

$E \in \mathcal{F}$ — any set function

Decompose into two parts:

$G_E \in \mathcal{D}$ "gross exclusion"

Exclusion and Reuse

$E \in \mathcal{F}$ - any set function

Decompose into two parts:

$G_E \in \mathcal{A}$ "gross exclusion"

$R_E \in \mathcal{C}$ "reuse"

Exclusion and Reuse

$E \in \mathcal{J}$ - any set function

Decompose into two parts:

$G_E \in \mathcal{D}$ "gross exclusion"

$R_E \in \mathcal{C}$ "reuse"

$\forall z \in [X]^*$

$$G_E(z) = E(z) \cup z$$

Exclusion and Reuse

$E \in \mathcal{J}$ – any set function

Decompose into two parts:

$G_E \in \mathcal{D}$ "gross exclusion"

$R_E \in \mathcal{C}$ "reuse"

$\forall z \in [X]^*$

$$G_E(z) = E(z) \cup z \qquad R_E(z) = z \setminus E(z)$$

Exclusion and Reuse

$E \in \mathcal{J}$ – any set function

Decompose into two parts:

$G_E \in \mathcal{D}$ "gross exclusion"

$R_E \in \mathcal{C}$ "reuse"

$\forall Z \in [X]^*$

$$G_E(Z) = E(Z) \cup Z \qquad R_E(Z) = Z \setminus E(Z)$$

$$\Rightarrow \quad E(Z) = G_E(Z) \setminus R_E(Z)$$

Exclusion and Reuse

Necessary conditions from "pure" cases
carry over

Exclusion and Reuse

Necessary conditions from "pure" cases carry over

Claim:

$$\left.\begin{array}{l} E \in \mathcal{J} \\ \Sigma_E \text{ preserves PI over } \mathcal{C}^{res} \end{array}\right\} \Longrightarrow \begin{array}{l} G_E \text{ is a } t\text{-exclusion} \\ R_E \text{ is a } \tilde{\tau}\text{-contraction} \end{array}$$

Exclusion and Reuse

Necessary conditions from "pure" cases carry over

Claim:

$$\left.\begin{array}{l} E \in \mathcal{S} \\ \Sigma_E \text{ preserves PI over } \mathcal{C}^{res} \end{array}\right\} \Longrightarrow \begin{array}{l} G_E \text{ is a } t\text{-exclusion} \\ R_E \text{ is a } \tilde{\tau}\text{-contraction} \end{array}$$

Not sufficient though

Exclusion and Reuse

Account for interaction between the two

Exclusion and Reuse

Account for interaction between the two

Disjointness:

$\forall z \in [X]^*$

$$R_E(z) \cap E(\{\}) = \{\}$$

Exclusion and Reuse

Account for interaction between the two

Disjointness:

$\forall z \in [X]^*$

$$R_E(z) \cap E(\{\}) = \{\}$$

always excluded (by monotonicity)

Exclusion and Reuse

Account for interaction between the two

Disjointness:

$\forall z \in [X]^*$

$$R_E(z) \cap E(\{\}) = \{\}$$

always excluded (by monotonicity)

Never reuse something that is always excluded

Exclusion and Reuse

Account for interaction between the two

Disjointness:

$\forall z \in [X]^*$

$$R_E(z) \cap E(\{\}) = \{\}$$

Claim:

$$\left.\begin{array}{l} E \in \mathcal{S} \\ \Sigma_E \text{ preserves PI over } \mathcal{E}^{res} \end{array}\right\} \Longrightarrow R_E \text{ is disjoint}$$

Exclusion and Reuse

G_E is a t-exclusion $\longrightarrow$ $t \in \mathbb{N} \cup \{0, \infty\}$

Exclusion and Reuse

G_E is a t-exclusion $\longrightarrow$ $t \in \mathbb{N} \cup \{0, \infty\}$

R_E is a $\mathcal{T}$-contraction $\longrightarrow$ $\{\} = T^0 \subseteq T^1 \subseteq T^2 \subseteq \cdots \subseteq T^t$

Exclusion and Reuse

G_E is a t-exclusion $\longrightarrow$ $t \in \mathbb{N} \cup \{0, \infty\}$

R_E is a $\mathcal{T}$-contraction $\longrightarrow$ $\{\} = T^0 \subseteq T^1 \subseteq T^2 \subseteq \dots \subseteq T^t$

R_E disjoint $\longrightarrow$ $E(\{\}) = K \subseteq X$ $\quad \subseteq X \setminus K$

Exclusion and Reuse

G_E is a t-exclusion $\longrightarrow$ $t \in \mathbb{N} \cup \{0, \infty\}$

R_E is a τ-contraction $\longrightarrow$ $\{\} = T^0 \subseteq T^1 \subseteq T^2 \subseteq \dots \subseteq T^t$

R_E disjoint $\longrightarrow$ $E(\{\}) = K \subseteq X$ $\qquad \subseteq X \setminus K$

$\forall z \in [X]^*$

$|z| < t \implies E(z) = \left(z \setminus T^{|z|}\right) \cup K$

$|z| \geq t \implies G_E(z) = X$

Exclusion and Reuse

G_E is a t-exclusion $\longrightarrow$ $t \in \mathbb{N} \cup \{0, \infty\}$

R_E is a Υ-contraction $\longrightarrow$ $\{\} = T^0 \subseteq T^1 \subseteq T^2 \subseteq \ldots \subseteq T^t$

R_E disjoint $\longrightarrow$ $E(\{\}) = K \subseteq X$ $\qquad \subseteq X \setminus K$

$\forall z \in [X]^*$

$|z| < t \implies E(z) = \left(z \setminus T^{|z|}\right) \cup K$

$|z| \geq t \implies G_E(z) = X$

E is a (t, K, Υ)-exclusion

Exclusion and Reuse

Theorem:

Σ_E preserves PI over $\mathcal{C}^{res}$ $\iff$ E is a (t, K, Υ)-exclusion

Extending This Result

Σ_E preserves PI over $\mathcal{C}^{res}$ $\iff$ E is a (t, K, γ)-exclusion

Extending This Result

Σ_E preserves PI over $\mathcal{C}^{PI}$ $\Longrightarrow$ E is a (t, k, γ)-exclusion

Necessity holds since $\mathcal{C}^{res} \subseteq \mathcal{C}^{PI}$

Extending This Result

Σ_E preserves PI over $\mathcal{C}^{PI}$ $\Longrightarrow$ E is a (t, k, γ)-exclusion

Sufficiency $\longrightarrow$ restrictions on parameters

Extending This Result

Σ_E preserves PI over $\mathcal{C}^{PI}$ $\Longrightarrow$ E is a (t, K, Υ)-exclusion

$K \subseteq X$

$t \in \mathbb{N} \cup \{0, \infty\}$

$$\{\} = T^0 \subseteq T^1 \subseteq T^2 \subseteq \ldots \subseteq T^t \subseteq X \setminus K$$

Extending This Result

Proposition:

Σ_E preserves PI over $\mathcal{C}^{PI}$ $\iff$ E is a (t, K, Υ)-exclusion, $K \subseteq X$, $t \in \{1\} \cup \{0, \infty\}$

$\{\} = T^0 \subseteq T^1 \subseteq T^2 = T^3 \ldots = T^t \subseteq X \setminus K$

Extending This Result

Another way to strengthen the left side:

Σ_E preserves PI and property π over $\mathcal{C}^{res}$ $\Longleftrightarrow$ E is a (t, k, γ)-exclusion

$+$

$\vdots$

Another Property

Size monotonicity (Alkan 2002, Alkan & Gale 2003, Fleiner 2003)

Another Property

Size monotonicity (Alkan 2002, Alkan & Gale 2003, Fleiner 2003)

Also called "law of aggregate demand" (Hatfield & Milgrom 2005)

Another Property

Size monotonicity (Alkan 2002, Alkan & Gale 2003, Fleiner 2003)

$$Y \subseteq Y' \Rightarrow |C(Y)| \leq |C(Y')|$$

Another Property

Size monotonicity (Alkan 2002, Alkan & Gale 2003, Fleiner 2003)

$$Y \subseteq Y' \Rightarrow |C(Y)| \leq |C(Y')|$$

- Comes up in the matching literature
  * ensures lattice structure of stable set
  * incentive compatibility of deferred acceptance

Another Property

Size monotonicity (Alkan 2002, Alkan & Gale 2003, Fleiner 2003)

$$Y \subseteq Y' \implies |C(Y)| \leq |C(Y')|$$

- Comes up in the matching literature
  - ensures lattice structure of stable set
  - incentive compatibility of deferred acceptance
- Responsiveness $\implies$ PI + SM

Extending This Result

Another way to strengthen the left side:

Σ_E preserves PI and SM over $\mathcal{C}^{res}$ $\iff$ E is a (t, k, γ)-exclusion

+

:

Extending This Result

Another way to strengthen the left side:

Σ_E preserves PI and SM over $\mathcal{C}^{res}$ $\iff$ E is a (t, k, Υ)-exclusion

$|X \setminus k| \leq 1$

or

$t = \infty$ and $\forall n$, $T^n = \{\}$

Extending This Result

Another way to strengthen the left side:

Σ_E preserves PI and SM over $\mathcal{C}^{PI+SM}$ $\iff$ E is a (t, K, Υ)-exclusion

$|X \setminus K| \leq 1$

or

$t = \infty$ and $\forall n,\ \Upsilon^n = \{\}$

Extending This Result

Another way to strengthen the left side:

Σ_E preserves PI and SM over $\mathcal{C}^{PI+SM}$ $\iff$ E is a (t, K, Υ)-exclusion

$|X \setminus K| \leq 1$

or

$t = \infty$ and $\forall n,\ \Upsilon^n = \{\}$

Both expanding the domain and preserving an extra property shrink the set of exclusion functions

Thanks!