Slow and Easy: a Theory of Browsing

ASSA, San Antonio, Texas, 2024

Evgenii Safonov, Queen Mary University of London

Motivation

- Window shopping/browsing
  - ✓ Not urgent
  - ✓ Attention may jump from item to item
  - ✓ Not known what options are available
  - ✓ Multiple attributes

Motivation

- ▷ Window shopping/browsing
  - ✓ Not urgent
  - ✓ Attention may jump from item to item
  - ✓ Not known what options are available
  - ✓ Multiple attributes
- ▷ "Classical" search models: cost of the search
  - ✓ Direct: information cost
  - ✓ Indirect: waiting cost

Motivation

- ▷ Window shopping/browsing
  - ✓ Not urgent
  - ✓ Attention may jump from item to item
  - ✓ Not known what options are available
  - ✓ Multiple attributes
- ▷ "Classical" search models: cost of the search
  - ✓ Direct: information cost
  - ✓ Indirect: waiting cost
- ▷ This paper: information processing constraints

Attributes

- ▷ Example—new TV

TV-set	technology	sound	brand	screen
a	OLED	excellent	S	50"
b	OLED	good	P	50"
c	LED	excellent	P	50"
d	LED	good	S	42"

Attributes

- ▷ Example—new TV

TV-set	technology	sound	brand	screen
a	OLED	excellent	S	50"
b	OLED	good	P	50"
c	LED	excellent	P	50"
d	LED	good	S	42"

- ▷ "Languages:"
  - ✓ Technology, sound quality

Attributes

- Example—new TV

TV-set	technology	sound	brand	screen
a	OLED	excellent	S	50”
b	OLED	good	P	50”
c	LED	excellent	P	50”
d	LED	good	S	42”

- “Languages:”
  - ✓ Technology, sound quality
  - ✓ Technology, brand

This Paper

- ▷ Objectives:
  - ✓ Find when the consumer can overcome the information processing constraints and achieve optimality if she takes her time
  - ✓ Identify descriptions of items that make the search easy

This Paper

- Objectives:
  - ✓ Find when the consumer can overcome the information processing constraints and achieve optimality if she takes her time
  - ✓ Identify descriptions of items that make the search easy
- Approach:
  - ✓ Use Automata Theory—automaton strategy as a decision making procedure

This Paper

- Objectives:
  - ✓ Find when the consumer can overcome the information processing constraints and achieve optimality if she takes her time
  - ✓ Identify descriptions of items that make the search easy
- Approach:
  - ✓ Use Automata Theory—automaton strategy as a decision making procedure
  - ✓ Consider zero waiting costs

This Paper

- ▷ Objectives:
  - ✓ Find when the consumer can overcome the information processing constraints and achieve optimality if she takes her time
  - ✓ Identify descriptions of items that make the search easy
- ▷ Approach:
  - ✓ Use Automata Theory—automaton strategy as a decision making procedure
  - ✓ Consider zero waiting costs
- ▷ Main insights:

This Paper

- Objectives:
  - ✓ Find when the consumer can overcome the information processing constraints and achieve optimality if she takes her time
  - ✓ Identify descriptions of items that make the search easy
- Approach:
  - ✓ Use Automata Theory—automaton strategy as a decision making procedure
  - ✓ Consider zero waiting costs
- Main insights:
  - ✓ Consumer choice is not hard if use randomization and sacrifice the speed: logarithmic/linear complexity

This Paper

- ▷ Objectives:
  - ✓ Find when the consumer can overcome the information processing constraints and achieve optimality if she takes her time
  - ✓ Identify descriptions of items that make the search easy
- ▷ Approach:
  - ✓ Use Automata Theory—automaton strategy as a decision making procedure
  - ✓ Consider zero waiting costs
- ▷ Main insights:
  - ✓ Consumer choice is not hard if use randomization and sacrifice the speed: logarithmic/linear complexity
  - ✓ System of attributes used to describe objects matters: languages that attain logarithmic/linear bounds in complexity

This Paper

- ▷ Objectives:
  - ✓ Find when the consumer can overcome the information processing constraints and achieve optimality if she takes her time
  - ✓ Identify descriptions of items that make the search easy
- ▷ Approach:
  - ✓ Use Automata Theory—automaton strategy as a decision making procedure
  - ✓ Consider zero waiting costs
- ▷ Main insights:
  - ✓ Consumer choice is not hard if use randomization and sacrifice the speed: logarithmic/linear complexity
  - ✓ System of attributes used to describe objects matters: languages that attain logarithmic/linear bounds in complexity
  - ✓ Simplest procedure: examine attributes sequentially, dismiss the item with positive probability if the attribute's value is bad

Model

Alternatives, Preference, and Menu

- Finite set of items A with generic element a
- Complete and transitive non-trivial preference $\succeq$ on A

Alternatives, Preference, and Menu

- Finite set of items A with generic element a
- Complete and transitive non-trivial preference $\succeq$ on A
- Nature chooses a non-empty menu $B \subseteq A$, unknown to the agent

Search

- In period $t = 1$, a random item is drawn from the menu

Search

- ▷ In period $t = 1$, a random item is drawn from the menu
- ▷ Agent investigates the item by examining its attributes:
  - ✓ One attribute (agent's choice) in a period
  - ✓ Can choose the item and stop the search
  - ✓ Can continue to investigate the item
  - ✓ Can dismiss the item and draw a new one

Search

- ▷ In period $t = 1$, a random item is drawn from the menu
- ▷ Agent investigates the item by examining its attributes:
  - ✓ One attribute (agent's choice) in a period
  - ✓ Can choose the item and stop the search
  - ✓ Can continue to investigate the item
  - ✓ Can dismiss the item and draw a new one
- ▷ Each period during the search:
  - ✓ With probability $\eta \in (0, 1)$, a new item catches the agent's attention
  - ✓ With probability $1 - \eta$, the item remains the same

Search

- ▷ In period $t = 1$, a random item is drawn from the menu
- ▷ Agent investigates the item by examining its attributes:
  - ✓ One attribute (agent's choice) in a period
  - ✓ Can choose the item and stop the search
  - ✓ Can continue to investigate the item
  - ✓ Can dismiss the item and draw a new one
- ▷ Each period during the search:
  - ✓ With probability $\eta \in (0, 1)$, a new item catches the agent's attention
  - ✓ With probability $1 - \eta$, the item remains the same
- ▷ Each time a random item is drawn according to the same distribution
  - ✓ Can encounter the same (or identical) item multiple times

Information Structures

- ▷ Language $Q = \{Q_i\}_{i \in N}$ is a collection of non-trivial binary partitions of A
  - ✓ Each partition maps to a binary property (attribute) of items
  - ✓ N—index set of partitions (attributes)
  - ✓ $a_i \in \{0, 1\}$ is the value of attribute $i \in N$

Information Structures

- ▷ Language $Q = \{Q_i\}_{i \in N}$ is a collection of non-trivial binary partitions of A
  - ✓ Each partition maps to a binary property (attribute) of items
  - ✓ N—index set of partitions (attributes)
  - ✓ $a_i \in \{0, 1\}$ is the value of attribute $i \in N$

TV-set	technology	sound	brand	screen
a	OLED	excellent	S	50”
b	OLED	good	P	50”
c	LED	excellent	P	50”
d	LED	good	S	42”

- ▷ Example: $\{Q_1, Q_2\}$, where $Q_1 = \{\{a, b\}, \{c, d\}\}$, $Q_2 = \{\{a, c\}, \{b, d\}\}$
  - ✓ Language includes *“technology”* and *“sound”* attributes

An Automaton Strategy

- Strategy (S, ι, τ)

An Automaton Strategy

- Strategy (S, ι, τ)
- State space $S = S^o \cup \{choose\} \cup \{dismiss\}$
  - ✓ $S^o = \{1, ..., m\}$—memory states
  - ✓ $\{choose, dismiss\}$—special states

An Automaton Strategy

- Strategy (S, ι, τ)
- State space $S = S^o \cup \{choose\} \cup \{dismiss\}$
  - ✓ $S^o = \{1, ..., m\}$—memory states
  - ✓ $\{choose, dismiss\}$—special states
- Interrogation rule $\iota : S^o \to N$
  - ✓ $\iota(s)$—attribute of the item investigated in state s

An Automaton Strategy

- Strategy (S, ι, τ)
- State space $S = S^o \cup \{choose\} \cup \{dismiss\}$
  - ✓ $S^o = \{1, ..., m\}$—memory states
  - ✓ $\{choose, dismiss\}$—special states
- Interrogation rule $\iota : S^o \to N$
  - ✓ $\iota(s)$—attribute of the item investigated in state s
- Stochastic transition rule $\tau : S^o \times \{0, 1\} \to \triangle(S)$
  - ✓ $\tau(s, v, j)$—probability to transition from s to v if attribute $\iota(s)$ has value j

An Automaton Strategy

- Strategy (S, ι, τ)
- State space $S = S^o \cup \{choose\} \cup \{dismiss\}$
  - ✓ $S^o = \{1, ..., m\}$—memory states
  - ✓ $\{choose, dismiss\}$—special states
- Interrogation rule $\iota : S^o \to N$
  - ✓ $\iota(s)$—attribute of the item investigated in state s
- Stochastic transition rule $\tau : S^o \times \{0, 1\} \to \triangle(S)$
  - ✓ $\tau(s, v, j)$—probability to transition from s to v if attribute $\iota(s)$ has value j
- Each time a new alternative is drawn, state initializes at $s = 1$
  - ✓ Agent focuses on the current item, no recall of the past investigations
  - ✓ In the paper, we relax this assumption for part of the analysis

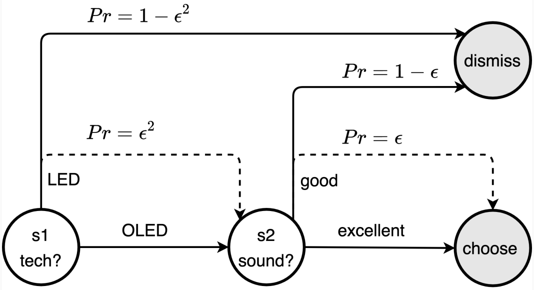
▸ Formal Dynamics

Example

- ▷ TV-set example, language: $\{technology, sound\}$
- ▷ Utility: $u(tech, sound) = 2 \cdot \mathbb{1}\{tech = OLED\} + 1 \cdot \mathbb{1}\{sound = excellent\}$

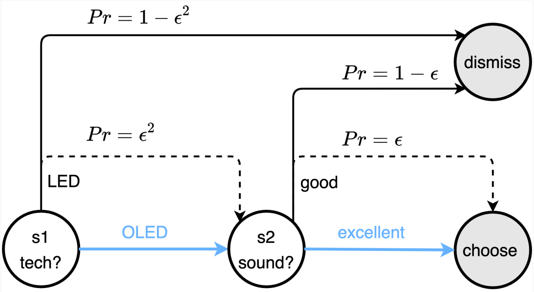
Example

- ▷ TV-set example, language: $\{technology, sound\}$
- ▷ Utility: $u(tech, sound) = 2 \cdot \mathbb{1}\{tech = OLED\} + 1 \cdot \mathbb{1}\{sound = excellent\}$



Example

- TV-set example, language: $\{technology, sound\}$
- Utility: $u(tech, sound) = 2 \cdot \mathbb{1}\{tech = OLED\} + 1 \cdot \mathbb{1}\{sound = excellent\}$

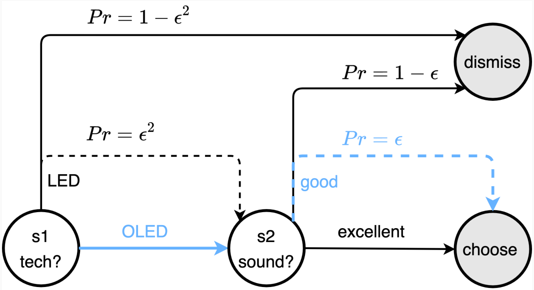


- Probability of choosing an item during an investigation:

$$q(OLED, excellent) = 1 - \eta \qquad q(OLED, good) = (1-\eta)\cdot\epsilon$$
$$q(LED, excellent) = (1-\eta)\cdot\epsilon^2 \qquad q(LED, good) = (1-\eta)\cdot\epsilon^3$$

Example

- TV-set example, language: $\{technology, sound\}$
- Utility: $u(tech, sound) = 2 \cdot \mathbb{1}\{tech = OLED\} + 1 \cdot \mathbb{1}\{sound = excellent\}$



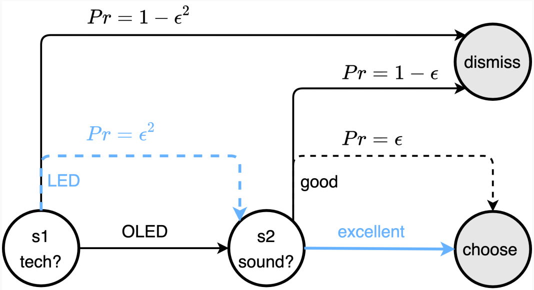
- Probability of choosing an item during an investigation:

$$q(OLED, excellent) = 1 - \eta \qquad q(OLED, good) = (1-\eta) \cdot \epsilon$$
$$q(LED, excellent) = (1-\eta) \cdot \epsilon^2 \qquad q(LED, good) = (1-\eta) \cdot \epsilon^3$$

Example

- TV-set example, language: $\{technology, sound\}$

- Utility: $u(tech, sound) = 2 \cdot \mathbb{1}\{tech = OLED\} + 1 \cdot \mathbb{1}\{sound = excellent\}$

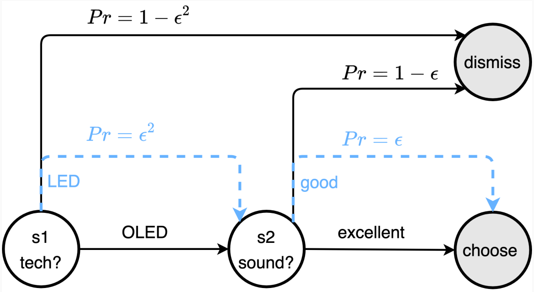


- Probability of choosing an item during an investigation:

$$
\begin{aligned}
q(OLED, excellent) &= 1 - \eta & \quad q(OLED, good) &= (1-\eta)\cdot\epsilon \\
q(LED, excellent) &= (1-\eta)\cdot\epsilon^2 & \quad q(LED, good) &= (1-\eta)\cdot\epsilon^3
\end{aligned}
$$

Example

- TV-set example, language: $\{technology, sound\}$
- Utility: $u(tech, sound) = 2 \cdot \mathbb{1}\{tech = OLED\} + 1 \cdot \mathbb{1}\{sound = excellent\}$

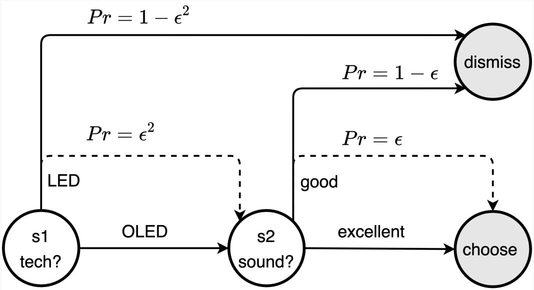


- Probability of choosing an item during an investigation:

$$q(OLED, excellent) = 1 - \eta \qquad q(OLED, good) = (1-\eta)\cdot\epsilon$$
$$q(LED, excellent) = (1-\eta)\cdot\epsilon^2 \qquad q(LED, good) = (1-\eta)\cdot\epsilon^3$$

Example

- TV-set example, language: $\{technology, sound\}$
- Utility: $u(tech, sound) = 2 \cdot \mathbb{1}\{tech = OLED\} + 1 \cdot \mathbb{1}\{sound = excellent\}$



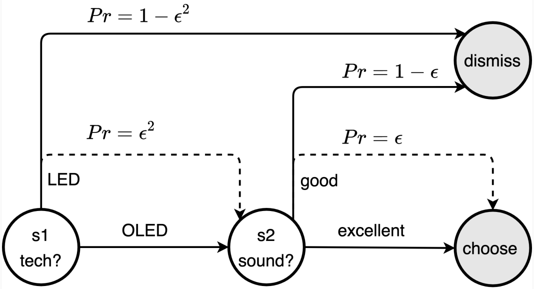
- Probability of choosing an item during an investigation:

$$q(OLED, excellent) = 1 - \eta \qquad q(OLED, good) = (1-\eta)\cdot\epsilon$$
$$q(LED, excellent) = (1-\eta)\cdot\epsilon^2 \qquad q(LED, good) = (1-\eta)\cdot\epsilon^3$$

- Imagine, the realized menu includes all but the best TV-set

Example

- ▷ TV-set example, language: $\{technology, sound\}$
- ▷ Utility: $u(tech, sound) = 2 \cdot \mathbb{1}\{tech = OLED\} + 1 \cdot \mathbb{1}\{sound = excellent\}$



- ▷ Probability of choosing an item during an investigation:

$$q(OLED, excellent) = 1 - \eta \qquad q(OLED, good) = (1-\eta)\cdot\epsilon$$
$$q(LED, excellent) = (1-\eta)\cdot\epsilon^2 \qquad q(LED, good) = (1-\eta)\cdot\epsilon^3$$

- ▷ $\epsilon \longrightarrow 0$, optimal choice from any menu with probability 1

▸ Formal Stochastic Choice

Solution Concept

- Choice problem $(Q, \succeq)$

Solution Concept

- Choice problem $(Q, \succeq)$
- Transitions of the automaton: $\mathcal{T} = \{(s, v, j) \mid \tau(s, v, j) > 0\}$

Solution Concept

- ▷ Choice problem $(Q, \succeq)$
- ▷ Transitions of the automaton: $\mathcal{T} = \big\{(s, v, j) \,\big|\, \tau(s, v, j) > 0\big\}$
- ▷ Decision rule $\psi = \{(S, \iota, \tau_r)\}_{r=1,2,\ldots}$
  - ✓ Fixed state space S
  - ✓ Fixed interrogation rule ι
  - ✓ Fixed transitions $\mathcal{T}_r = \mathcal{T}$, $r = 1, 2, \ldots$

Solution Concept

- ▷ Choice problem $(Q, \succeq)$
- ▷ Transitions of the automaton: $\mathcal{T} = \{(s, v, j) \mid \tau(s, v, j) > 0\}$
- ▷ Decision rule $\psi = \{(S, \iota, \tau_r)\}_{r=1,2,\ldots}$
  - ✓ Fixed state space S
  - ✓ Fixed interrogation rule ι
  - ✓ Fixed transitions $\mathcal{T}_r = \mathcal{T}$, $r = 1, 2, \ldots$

Definition. *A decision rule ψ solves the choice problem $(Q, \succeq)$ if*

$$\lim_{r \to \infty} \Pr\big(\text{choose } \succeq \text{-best item from menu } B\big) = 1 \qquad \forall B$$

Existence of a Solution

Proposition. *There exists a decision rule that solves the agent's choice problem if and only for any* $a, b \in A$*, if* $a \succ b$*, then* $a_i \neq b_i$ *for some* $i \in N$*.*

Existence of a Solution

Proposition. *There exists a decision rule that solves the agent's choice problem if and only for any $a, b \in A$, if $a \succ b$, then $a_i \neq b_i$ for some $i \in N$.*

- ▷ We consider languages that allow the agent to solve her choice problem

Existence of a Solution

Proposition. *There exists a decision rule that solves the agent's choice problem if and only for any* $a, b \in A$*, if* $a \succ b$*, then* $a_i \neq b_i$ *for some* $i \in N$*.*

- We consider languages that allow the agent to solve her choice problem
- Given the agent's language, what is the minimum amount of cognitive resources required to solve the choice problem?

Complexity Measures

In the paper:

- ▷ Memory load of a decision rule: $\mathcal{M}(\psi) = |S^o|$
  - ✓ Represents an "operational" memory required to implement the procedure
- ▷ Memory load of a language (given $\succeq$):

$$\mathcal{M}_{\succeq}(Q) := \min_{\psi \text{ solves } (Q,\succeq)} \mathcal{M}(\psi)$$

Complexity Measures

In the paper:

- ▷ Memory load of a decision rule: $\mathcal{M}(\psi) = |S^o|$
  - ✓ Represents an "operational" memory required to implement the procedure
- ▷ Memory load of a language (given $\succeq$):

$$\mathcal{M}_{\succeq}(Q) := \min_{\psi \text{ solves } (Q,\succeq)} \mathcal{M}(\psi)$$

This talk:

- ▷ Complexity of a decision rule: $\kappa(\psi) = |\{(s, v, j) \mid \tau(s, v, j) > 0\}|$

Complexity Measures

In the paper:

- ▷ Memory load of a decision rule: $\mathcal{M}(\psi) = |S^o|$
  - ✓ Represents an "operational" memory required to implement the procedure
- ▷ Memory load of a language (given $\succeq$):

$$\mathcal{M}_{\succeq}(Q) := \min_{\psi \text{ solves } (Q,\succeq)} \mathcal{M}(\psi)$$

This talk:

- ▷ Complexity of a decision rule: $\kappa(\psi) = \left|\{(s,v,j) \mid \tau(s,v,j) > 0\}\right|$
  - ✓ Represents "length of instructions" of the procedure

Complexity Measures

In the paper:

- Memory load of a decision rule: $\mathcal{M}(\psi) = |S^o|$
  - ✓ Represents an "operational" memory required to implement the procedure
- Memory load of a language (given $\succeq$):

$$\mathcal{M}_{\succeq}(Q) := \min_{\psi \text{ solves } (Q,\succeq)} \mathcal{M}(\psi)$$

This talk:

- Complexity of a decision rule: $\kappa(\psi) = |\{(s,v,j) \mid \tau(s,v,j) > 0\}|$
  - ✓ Represents "length of instructions" of the procedure
  - ✓ Considered for repeated games in Banks and Sundaram (1990)

Complexity Measures

In the paper:

- Memory load of a decision rule: $\mathcal{M}(\psi) = |S^o|$
  - ✓ Represents an "operational" memory required to implement the procedure
- Memory load of a language (given $\succeq$):

$$\mathcal{M}_{\succeq}(Q) := \min_{\psi \text{ solves } (Q,\succeq)} \mathcal{M}(\psi)$$

This talk:

- Complexity of a decision rule: $\kappa(\psi) = |\{(s, v, j) \mid \tau(s, v, j) > 0\}|$
  - ✓ Represents "length of instructions" of the procedure
  - ✓ Considered for repeated games in Banks and Sundaram (1990)
  - ✓ Experimentally in Oprea (2020)

Complexity Measures

In the paper:

- Memory load of a decision rule: $\mathcal{M}(\psi) = |S^o|$
  - ✓ Represents an "operational" memory required to implement the procedure
- Memory load of a language (given $\succeq$):

$$\mathcal{M}_{\succeq}(Q) := \min_{\psi \text{ solves } (Q,\succeq)} \mathcal{M}(\psi)$$

This talk:

- Complexity of a decision rule: $\kappa(\psi) = |\{(s, v, j) \mid \tau(s, v, j) > 0\}|$
  - ✓ Represents "length of instructions" of the procedure
  - ✓ Considered for repeated games in Banks and Sundaram (1990)
  - ✓ Experimentally in Oprea (2020)
- Complexity (transitional) of a language (given $\succeq$):

$$\kappa_{\succeq}(Q) := \min_{\psi \text{ solves } (Q,\succeq)} \kappa(\psi)$$

Complexity of Languages for 4 Items and Strict Preference

Consider $A = \{a, b, c, d\}$, and $a \succ b \succ c \succ d$

	Language	Preference	$\mathcal{M}$	κ
Q	$\{\{a,b\},\{c,d\}\},\ \{\{a,c\},\{b,d\}\}$	$11 \succ 10 \succ 01 \succ 00$	2	6
R	$\{\{a,b\},\{c,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 10 \succ 00 \succ 01$	2	7
S	$\{\{a,c\},\{b,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 00 \succ 10 \succ 01$	3	9
T	$\{\{a\},\{b,c,d\}\},\ \{\{b\},\{a,c,d\}\},\ \{\{c\},\{a,b,d\}\}$	$100 \succ 010 \succ 001 \succ 000$	3	9

Some details

Maximum Complexity

Theorem (Upper Bound). *If there are $k = |A|$ items, then for any $\succeq$:*

(i) For any language Q, $\kappa_{\succeq}(Q) \leq 3k - 3$;

(ii) There exists a language Q such that $\kappa_{\succeq}(Q) \geq k - 2$.

▸ Proof Idea for (i)

Minimum Complexity

Theorem (Lower Bound). *Let $\succeq$ have m indifference classes, then:*

(i) For any language Q, $\kappa_{\succeq}(Q) \geq 3\lceil \log_2 m \rceil$;

(ii) There exists a language Q such that $\kappa_{\succeq}(Q) = 3\lceil \log_2 m \rceil$;

(iii) If ψ *solves* $(Q, \succeq)$, *and* $\kappa(\psi) = 3\lceil \log_2 m \rceil$, *then* $\mathcal{M}(\psi)$ *is minimum among the rules that solve the choice problem* $(\widetilde{Q}, \succeq)$ *for any language* $\widetilde{Q}$,

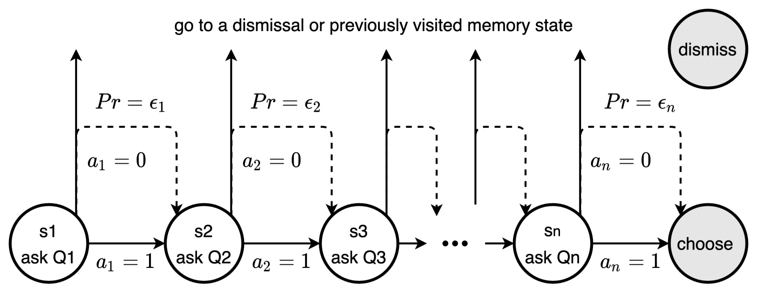
where $\lceil x \rceil$ *denotes the smallest natural number weakly greater than x.*

▸ Proof Idea for (i)

Simplest Languages and Decision Rules

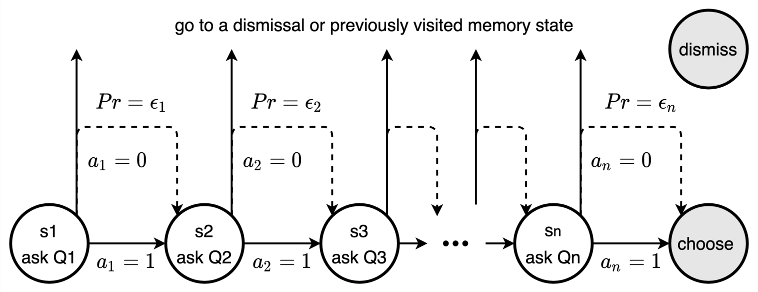
Separable Decision Rules

▷ Consider the following decision rules with $\{\epsilon_i\}_{r=1,2,\ldots} \in (0,1)$



Separable Decision Rules

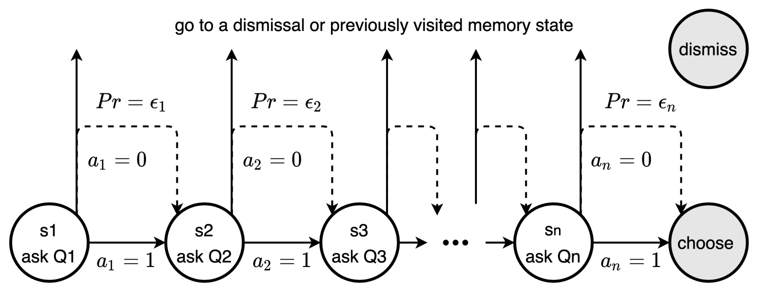
- Consider the following decision rules with $\{\epsilon_i\}_{r=1,2,...} \in (0,1)$



- Can enumerate attributes and attributes' values arbitrarily

Separable Decision Rules

- ▷ Consider the following decision rules with $\{\epsilon_i\}_{r=1,2,\ldots} \in (0,1)$



- ▷ Can enumerate attributes and attributes' values arbitrarily
- ▷ Call Ψ_n^+ the set of such rules with n memory states

Additive Utility

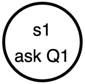
- Suppose the agent's language facilitates usage of an additive utility:

$$a \succ b \quad \Longrightarrow \quad \sum_{i \in N} \lambda_i a_i > \sum_{i \in N} \lambda_i b_i, \qquad (WLOG)\ \lambda_i \geq 0$$

Additive Utility

- Suppose the agent's language facilitates usage of an additive utility:

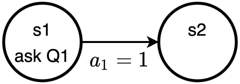
$$a \succ b \implies \sum_{i \in N} \lambda_i a_i > \sum_{i \in N} \lambda_i b_i, \qquad (WLOG)\ \lambda_i \geq 0$$



Additive Utility

- Suppose the agent's language facilitates usage of an additive utility:

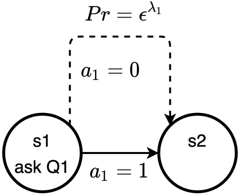
$$a \succ b \implies \sum_{i \in N} \lambda_i a_i > \sum_{i \in N} \lambda_i b_i, \quad (WLOG)\ \lambda_i \geq 0$$



Additive Utility

- Suppose the agent's language facilitates usage of an additive utility:

$$a \succ b \implies \sum_{i \in N} \lambda_i a_i > \sum_{i \in N} \lambda_i b_i, \qquad (WLOG)\ \lambda_i \geq 0$$



Additive Utility

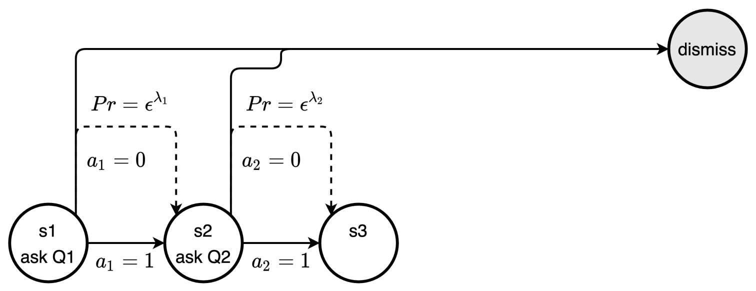
- Suppose the agent's language facilitates usage of an additive utility:

$$a \succ b \implies \sum_{i \in N} \lambda_i a_i > \sum_{i \in N} \lambda_i b_i, \quad (WLOG)\ \lambda_i \geq 0$$

Additive Utility

- Suppose the agent's language facilitates usage of an additive utility:

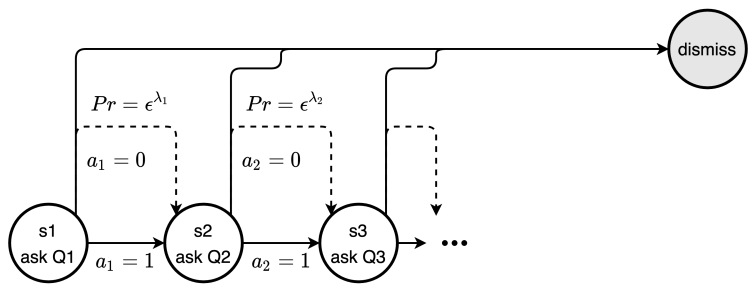
$$a \succ b \quad \Longrightarrow \quad \sum_{i \in N} \lambda_i a_i > \sum_{i \in N} \lambda_i b_i, \qquad (WLOG)\ \lambda_i \geq 0$$



Additive Utility

▷ Suppose the agent's language facilitates usage of an additive utility:

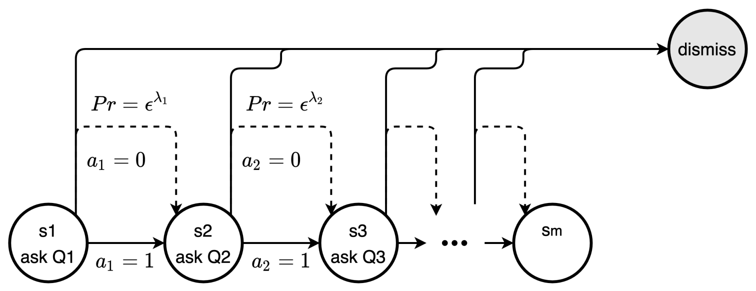
$$a \succ b \implies \sum_{i \in N} \lambda_i a_i > \sum_{i \in N} \lambda_i b_i, \qquad (WLOG)\ \lambda_i \geq 0$$



Additive Utility

▷ Suppose the agent's language facilitates usage of an additive utility:

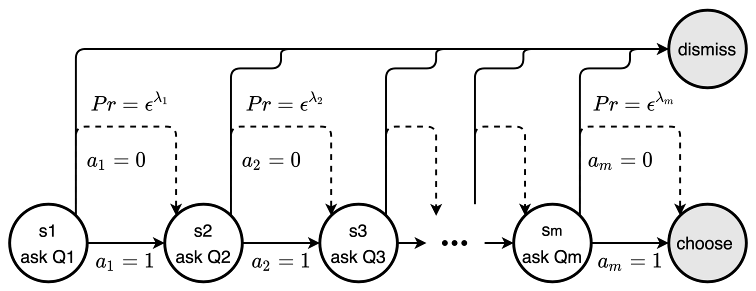
$$a \succ b \implies \sum_{i \in N} \lambda_i a_i > \sum_{i \in N} \lambda_i b_i, \quad (WLOG)\ \lambda_i \geq 0$$



Additive Utility

▷ Suppose the agent's language facilitates usage of an additive utility:

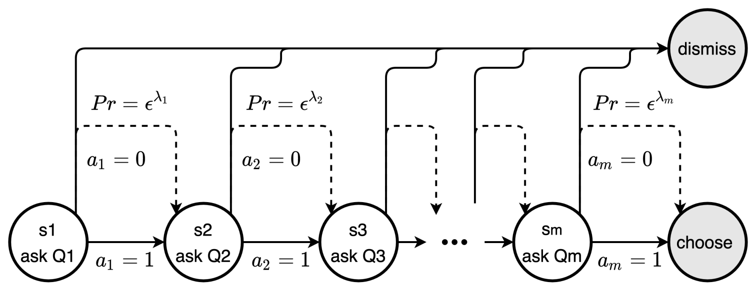
$$a \succ b \implies \sum_{i \in N} \lambda_i a_i > \sum_{i \in N} \lambda_i b_i, \qquad (WLOG)\ \lambda_i \geq 0$$



Additive Utility

- Suppose the agent's language facilitates usage of an additive utility:

$$a \succ b \quad \Longrightarrow \quad \sum_{i \in N} \lambda_i a_i > \sum_{i \in N} \lambda_i b_i, \qquad (WLOG)\ \lambda_i \geq 0$$



- Pr(choose item a during single investigation)$= (1-\eta)^{m-1} \cdot \epsilon^{\sum \lambda_i (1-a_i)}$

Adapted Languages

Definition. *Let $\succeq$ have m indifference classes. Language Q is adapted for $\succeq$ if there exists $\lambda \in \mathbb{R}^N$ such that:*

$$
\begin{array}{ll}
(i) & a \succ b \implies \sum_{i \in N} \lambda_i a_i > \sum_{i \in N} \lambda_i b_i \\
(ii) & \left|\{i \in N | \lambda_i \neq 0\}\right| = \lceil \log_2 m \rceil
\end{array}
$$

Adapted Languages

Definition. *Let $\succeq$ have m indifference classes. Language Q is adapted for $\succeq$ if there exists $\lambda \in \mathbb{R}^N$ such that:*

$$(i) \quad a \succ b \quad \implies \quad \sum_{i \in N} \lambda_i a_i > \sum_{i \in N} \lambda_i b_i$$

$$(ii) \quad \left|\{i \in N | \lambda_i \neq 0\}\right| \; = \; \lceil \log_2 m \rceil$$

Proposition. *There exists an adapted language.*

▸ Proof

Adapted Languages

Definition. *Let $\succeq$ have m indifference classes. Language Q is adapted for $\succeq$ if there exists $\lambda \in \mathbb{R}^N$ such that:*

$$(i) \quad a \succ b \quad \implies \quad \sum_{i \in N} \lambda_i a_i > \sum_{i \in N} \lambda_i b_i$$

$$(ii) \quad \left|\{i \in N | \lambda_i \neq 0\}\right| \; = \; \lceil \log_2 m \rceil$$

Proposition. *There exists an adapted language.*

▸ Proof

Remark. The utility function $u(a) = \sum_{i \in N} \lambda_i a_i$ induces a preference that might break ties in the original preference $\succeq$.

Simplest Decision Rules and Adapted Languages

Proposition. *Let $\succeq$ have m indifference classes, then Q is adapted for $\succeq$ if and only if there exists $\psi \in \Psi^{+}_{\lceil \log_2 m \rceil}$ that solves $(Q, \succeq)$.*

Simplest Languages

Theorem (Simplest Languages). *Let $\succeq$ have m indifference classes, then:*

(i) If Q is adapted for $\succeq$, then $\kappa_{\succeq}(Q) = 3\lceil \log_2 m \rceil$;

(ii) If $(3/4) \cdot 2^n < m \leq 2^n$ for a natural n, then:

(a) $\kappa_{\succeq}(Q) = 3\lceil \log_2 m \rceil$ if and only if Q is adapted for $\succeq$;

(b) If ψ solves $(Q, \succeq)$, and $\kappa(\psi) = 3\lceil \log_2 m \rceil$, then $\psi \in \Psi^+_{\lceil \log_2 m \rceil}$.

▸ Proof Sketch

Literature Review and Conclusion

Literature Review

- **Optimal search:** Kohn and Shavell (1974); Weitzman (1979); Morgan and Manning (1985); Klabjan, Olszewski, and Wolinsky (2014); Sanjurjo (2017)
- **Memory-constrained search:** Dow (1991); Sanjurjo (2015), (2019)
- **Stochastic Browsing:** Cerreia-Vioglio, Maccheroni, Marinacci, Rustichini (2020), Rustichini (2020)
- **Hypothesis testing and learning with finite memory:** Cover (1969); Cover and Hellman (1970); Hellman and Cover (1970), (1971)
- **Automata and simple algorithms in Economics:** Abreu and Rubinstein (1988); Kalai and Stanford (1988); Banks and Sundaram (1990); Kalai and Solan (2003); Börgers and Morales (2004); Kocer (2010); Salant (2011); Mandler, Manzini, Mariotti (2012); Wilson (2014); Oprea (2020)

Conclusion

- Simple stochastic strategies achieve near optimality when time is not of the essence
- Descriptions that facilitate additive utility with few attributes are key for simplicity
- In the simplest procedures, “higher” memory state indicate higher quality of the item relative to the menu

Supplementary Slides

Maximum and Minimum Memory Load

Theorem (Upper Bound). *If there are $k = |A|$ items, then for any $\succeq$:*

(i) For any language Q, $\kappa_{\succeq}(Q) \leq k - 1$;

(ii) There exists a language Q such that $\kappa_{\succeq}(Q) \geq k/2 - 1$.

Theorem (Lower Bound). *Let $\succeq$ have m indifference classes, then:*

(i) For any language Q, $\kappa_{\succeq}(Q) \geq \lceil \log_2 m \rceil$;

(ii) There exists a language Q such that $\kappa_{\succeq}(Q) = \lceil \log_2 m \rceil$;

where $\lceil x \rceil$ denotes the smallest natural number weakly greater than x.

Extension: Relaxing Memory Initialization Assumption

A General Framework

- Baseline model: a state initializes at $s = 1$ with each new item

A General Framework

- Baseline model: a state initializes at $s = 1$ with each new item
- General model: when a new item is drawn, the automaton transitions to a new state conditional on the previous state

A General Framework

- Baseline model: a state initializes at $s = 1$ with each new item
- General model: when a new item is drawn, the automaton transitions to a new state conditional on the previous state
- State space $S = S^o \cup \{choose\}$

A General Framework

- ▷ Baseline model: a state initializes at $s = 1$ with each new item
- ▷ General model: when a new item is drawn, the automaton transitions to a new state conditional on the previous state
- ▷ State space $S = S^o \cup \{choose\}$
- ▷ Specify probabilities:
  - ✓ To choose the current item, conditional on the current state and the learned attribute's value
  - ✓ To continue the investigation of the item and move to a memory state, conditional on the current state and the learned attribute's value
  - ✓ To dismiss the item, pick a new random item, and move to a memory state, conditional on the current state and the learned attribute's value
  - ✓ To move to a memory state, conditional on the current state and the event that a new item catches the agent's attention

Maximal Memory Load

Theorem (Upper Bound). *Consider a general model. Let k be the total number of items, then for any non-trivial $\succeq$:*

(i) For any language Q, $\mathcal{M}(Q) \leq k-1$;

(ii) There exists a language Q such that $\mathcal{M}(Q) = k/2 - 1$.

Minimal Memory Load

Theorem (Lower Bound). *Consider a general model. Let $m \geq 2$ be the total number of indifference classes of $\succeq$, then:*

(i) For any language Q, $\mathcal{M}(Q) \geq \lceil \log_2 m \rceil$;

(ii) There exists a language Q such that $\mathcal{M}(Q) = \lceil \log_2 m \rceil$.

If Preference is Strict, a Language May Require $k-1$ Memory States

- Let $A = \{a^1, ..., a^k\}$, $a^1 \succ ... \succ a^k$
- Consider $Q = \{Q_1, ..., Q_{k-1}\}$ with $Q_l = \{\{a^l\}, \{a^1, ..., a^{l-1}, a^{l+1}, ..., a^k\}\}$
- Need at least $k-1$ attributes to differentiate any pair of items

Proof Ideas

Lower Bound in Transitional Complexity—Simple Paths

- ▷ Focus on simple paths from $s = 1$ to $s = \textit{choose}$
- ▷ Item-dependent probability that the path occurs
- ▷ For $a \in A$, $\omega(a)$— the highest probability among all simple paths

Lemma. *A decision rule solves the choice problem if and only if:*

(i) $a \succ b$ *implies* $\omega(b)/\omega(a) \longrightarrow 0$ *for all* $a, b \in A$*;*

(ii) $\omega(a) > 0$ *for all* $a \in A$*.*

- ▷ Similar to "Z-tree" technique in Kandori, Mailath, Rob (1993)

Back

Strong and Weak Transitions

- ▷ Strong link $(s, v, j) \in \mathcal{T}$ if $\lim \tau(s, v, j) > 0$
- ▷ Weak link $(s, v, j) \in \mathcal{T}$ if $\lim \tau(s, v, j) = 0$

Lemma. *If the decision rule solves the choice problem, then highest-probability paths for different alternatives use different sets of weak links.*

Back

Lower Bound in Transitional Complexity—Proof Idea

- Let ψ solves $(Q, \succeq)$ with k items, $n = \lceil \log_2 k \rceil$
- ψ should have at least $2n$ strong links
  - ✓ At least n attributes should be examined in n states
  - ✓ Each state has at least 2 outgoing strong links
- ψ should have at least n weak links
  - ✓ Each item maps to a distinct set of weak links
  - ✓ Hence $2^{\#\text{weak links}} \geq k$
- The total number of links in ψ is at least $2n + n$, i.e. $\kappa(Q) \geq 3n$
- If $\kappa(\psi) = 3n$, there are exactly $2n$ strong and n weak links

Back

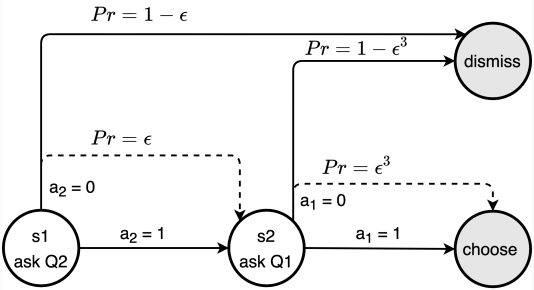
Memory Load—a Rough Complexity Measure

	Language	Preference	Memory load
Q	$\{\{a,b\},\{c,d\}\},\ \{\{a,c\},\{b,d\}\}$	$11 \succ 10 \succ 01 \succ 00$	$\mathcal{M}(Q) = 2$
Q^*	$\{\{a,b\},\{c,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 10 \succ 00 \succ 01$	$\mathcal{M}(Q^*) = 2$
Q^{**}	$\{\{a,c\},\{b,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 00 \succ 10 \succ 01$	$\mathcal{M}(Q^{**}) = 3$

Back

Memory Load—a Rough Complexity Measure

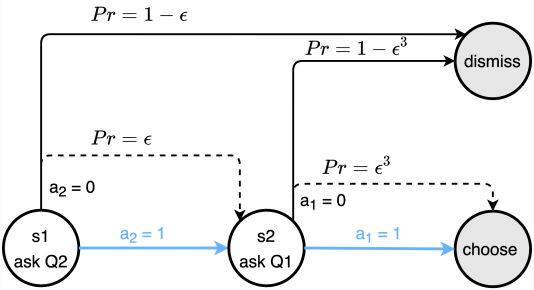
	Language	Preference	Memory load
Q	$\{\{a,b\},\{c,d\}\},\ \{\{a,c\},\{b,d\}\}$	$11 \succ 10 \succ 01 \succ 00$	$\mathcal{M}(Q) = 2$
Q^*	$\{\{a,b\},\{c,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 10 \succ 00 \succ 01$	$\mathcal{M}(Q^*) = 2$
Q^{**}	$\{\{a,c\},\{b,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 00 \succ 10 \succ 01$	$\mathcal{M}(Q^{**}) = 3$



Back

Memory Load—a Rough Complexity Measure

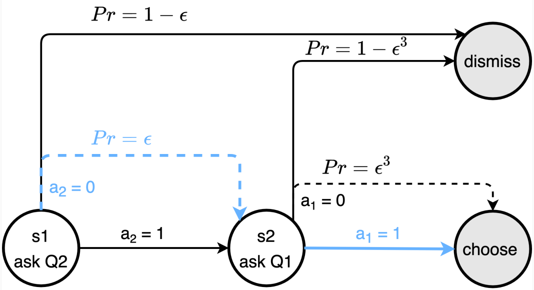
	Language	Preference	Memory load
Q	$\{\{a,b\},\{c,d\}\},\ \{\{a,c\},\{b,d\}\}$	$11 \succ 10 \succ 01 \succ 00$	$\mathcal{M}(Q) = 2$
Q^*	$\{\{a,b\},\{c,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 10 \succ 00 \succ 01$	$\mathcal{M}(Q^*) = 2$
Q^{**}	$\{\{a,c\},\{b,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 00 \succ 10 \succ 01$	$\mathcal{M}(Q^{**}) = 3$



▸ Back

Memory Load—a Rough Complexity Measure

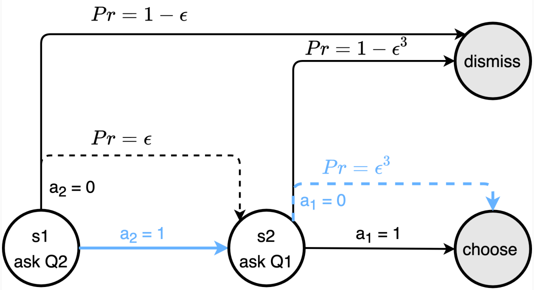
	Language	Preference	Memory load
Q	$\{\{a,b\},\{c,d\}\},\ \{\{a,c\},\{b,d\}\}$	$11 \succ 10 \succ 01 \succ 00$	$\mathcal{M}(Q) = 2$
Q^*	$\{\{a,b\},\{c,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 10 \succ 00 \succ 01$	$\mathcal{M}(Q^*) = 2$
Q^{**}	$\{\{a,c\},\{b,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 00 \succ 10 \succ 01$	$\mathcal{M}(Q^{**}) = 3$



▸ Back

Memory Load—a Rough Complexity Measure

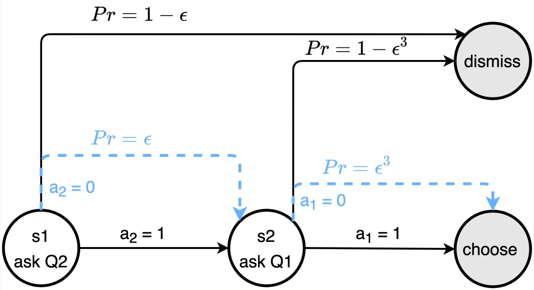
	Language	Preference	Memory load
Q	$\{\{a,b\},\{c,d\}\},\ \{\{a,c\},\{b,d\}\}$	$11 \succ 10 \succ 01 \succ 00$	$\mathcal{M}(Q) = 2$
Q^*	$\{\{a,b\},\{c,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 10 \succ 00 \succ 01$	$\mathcal{M}(Q^*) = 2$
Q^{**}	$\{\{a,c\},\{b,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 00 \succ 10 \succ 01$	$\mathcal{M}(Q^{**}) = 3$



Back

Memory Load—a Rough Complexity Measure

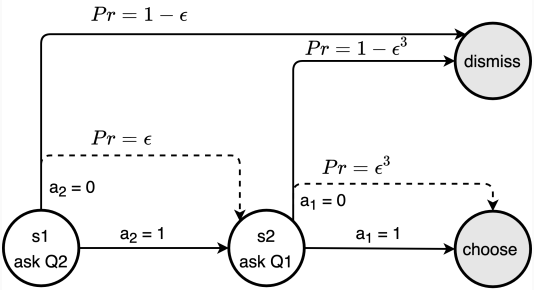
	Language	Preference	Memory load
Q	$\{\{a,b\},\{c,d\}\},\ \{\{a,c\},\{b,d\}\}$	$11 \succ 10 \succ 01 \succ 00$	$\mathcal{M}(Q) = 2$
Q^*	$\{\{a,b\},\{c,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 10 \succ 00 \succ 01$	$\mathcal{M}(Q^*) = 2$
Q^{**}	$\{\{a,c\},\{b,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 00 \succ 10 \succ 01$	$\mathcal{M}(Q^{**}) = 3$



Back

Memory Load—a Rough Complexity Measure

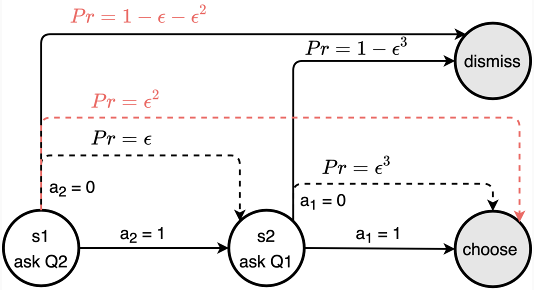
	Language	Preference	Memory load
Q	$\{\{a,b\},\{c,d\}\},\ \{\{a,c\},\{b,d\}\}$	$11 \succ 10 \succ 01 \succ 00$	$\mathcal{M}(Q) = 2$
Q^*	$\{\{a,b\},\{c,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 10 \succ 00 \succ 01$	$\mathcal{M}(Q^*) = 2$
Q^{**}	$\{\{a,c\},\{b,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 00 \succ 10 \succ 01$	$\mathcal{M}(Q^{**}) = 3$



Back

Memory Load—a Rough Complexity Measure

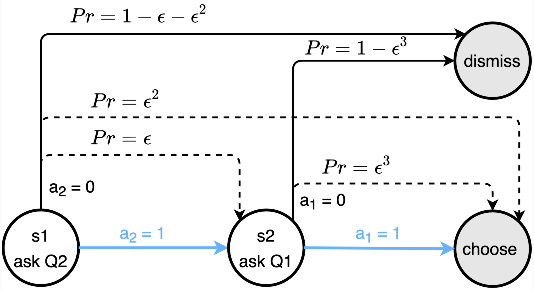
	Language	Preference	Memory load
Q	$\{\{a,b\},\{c,d\}\},\ \{\{a,c\},\{b,d\}\}$	$11 \succ 10 \succ 01 \succ 00$	$\mathcal{M}(Q) = 2$
Q^*	$\{\{a,b\},\{c,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 10 \succ 00 \succ 01$	$\mathcal{M}(Q^*) = 2$
Q^{**}	$\{\{a,c\},\{b,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 00 \succ 10 \succ 01$	$\mathcal{M}(Q^{**}) = 3$



▸ Back

Memory Load—a Rough Complexity Measure

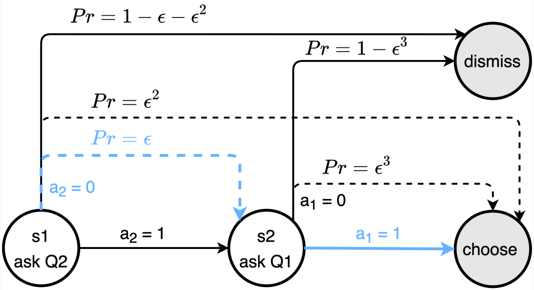
	Language	Preference	Memory load
Q	$\{\{a,b\},\{c,d\}\},\ \{\{a,c\},\{b,d\}\}$	$11 \succ 10 \succ 01 \succ 00$	$\mathcal{M}(Q) = 2$
Q^*	$\{\{a,b\},\{c,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 10 \succ 00 \succ 01$	$\mathcal{M}(Q^*) = 2$
Q^{**}	$\{\{a,c\},\{b,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 00 \succ 10 \succ 01$	$\mathcal{M}(Q^{**}) = 3$



Back

Memory Load—a Rough Complexity Measure

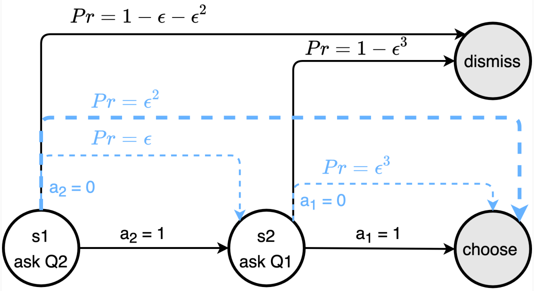
	Language	Preference	Memory load
Q	$\{\{a,b\},\{c,d\}\},\ \{\{a,c\},\{b,d\}\}$	$11 \succ 10 \succ 01 \succ 00$	$\mathcal{M}(Q) = 2$
Q^*	$\{\{a,b\},\{c,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 10 \succ 00 \succ 01$	$\mathcal{M}(Q^*) = 2$
Q^{**}	$\{\{a,c\},\{b,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 00 \succ 10 \succ 01$	$\mathcal{M}(Q^{**}) = 3$



▸ Back

Memory Load—a Rough Complexity Measure

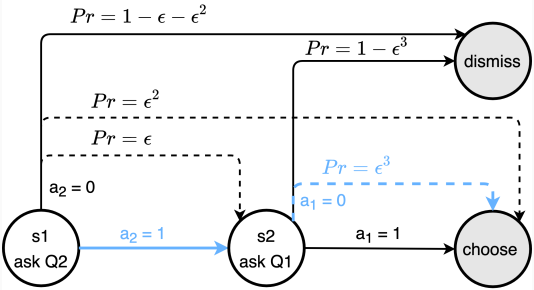
	Language	Preference	Memory load
Q	$\{\{a,b\},\{c,d\}\},\ \{\{a,c\},\{b,d\}\}$	$11 \succ 10 \succ 01 \succ 00$	$\mathcal{M}(Q) = 2$
Q^*	$\{\{a,b\},\{c,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 10 \succ 00 \succ 01$	$\mathcal{M}(Q^*) = 2$
Q^{**}	$\{\{a,c\},\{b,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 00 \succ 10 \succ 01$	$\mathcal{M}(Q^{**}) = 3$



▸ Back

Memory Load—a Rough Complexity Measure

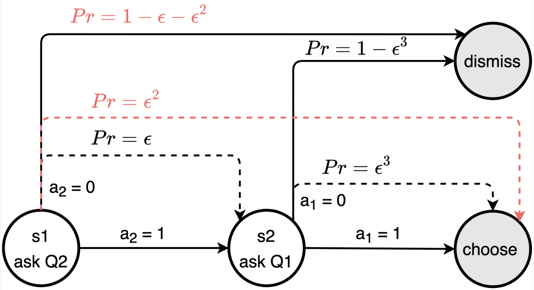
	Language	Preference	Memory load
Q	$\{\{a,b\},\{c,d\}\},\ \{\{a,c\},\{b,d\}\}$	$11 \succ 10 \succ 01 \succ 00$	$\mathcal{M}(Q) = 2$
Q^*	$\{\{a,b\},\{c,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 10 \succ 00 \succ 01$	$\mathcal{M}(Q^*) = 2$
Q^{**}	$\{\{a,c\},\{b,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 00 \succ 10 \succ 01$	$\mathcal{M}(Q^{**}) = 3$



▸ Back

Memory Load—a Rough Complexity Measure

	Language	Preference	Memory load
Q	$\{\{a,b\},\{c,d\}\},\ \{\{a,c\},\{b,d\}\}$	$11 \succ 10 \succ 01 \succ 00$	$\mathcal{M}(Q) = 2$
Q^*	$\{\{a,b\},\{c,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 10 \succ 00 \succ 01$	$\mathcal{M}(Q^*) = 2$
Q^{**}	$\{\{a,c\},\{b,d\}\},\ \{\{a,d\},\{b,c\}\}$	$11 \succ 00 \succ 10 \succ 01$	$\mathcal{M}(Q^{**}) = 3$



Back

Dynamics (Baseline Model)

- Markov Chain $\boldsymbol{Y} = (Y_1, Y_2, ...)$ with realizations $(y_1, y_2, ...)$
- Interpretation: $y_t = (a, s) \in A \times \left(S^o \cup \{choose\}\right)$
- Starting state: $Pr\left(Y_1 = (a, s)\right) = \rho^B(a) \cdot \delta_1^s$
- Transitional probabilities

$$
\begin{aligned}
Pr\Big(Y_t = (a,s) \;\Big|\; Y_{t-1} = (b,v)\Big) \quad &= \quad (1-\eta) \cdot \delta_b^a \cdot \tau\big(v, s, b_{\iota(v)}\big) + \\
& \quad\;\; (1-\eta) \cdot \tau\big(v, dismiss, b_{\iota(v)}\big) \cdot \rho^B(a) \cdot \delta_1^s + \\
& \quad\;\; \big[1 - \tau\big(v, choose, b_{\iota(v)}\big)\big] \cdot \eta \cdot \rho^B(a) \cdot \delta_1^s
\end{aligned}
$$

$$
Pr\Big(Y_t = (a, choose) \;\Big|\; Y_{t-1} = (b,v)\Big) \quad = \quad \tau\big(v, choose, b_{\iota(v)}\big) \cdot \delta_b^a
$$

$$
Pr\Big(Y_t = (a,s) \;\Big|\; Y_{t-1} = (b, choose)\Big) \quad = \quad \delta_b^a \cdot \delta_{choose}^s
$$

- Where $\rho^B(a)$ is the probability to draw item a from menu B

▸ Back

Stochastic Choice

- $\rho^B(b)$—probability to draw item b from menu B
- $q(b)$—probability to choose item b during a single investigation
- $p^B(b)$—probability to choose item b from menu B

Lemma (Generalized Luce Rule).

$$p^B(a) \;=\; \frac{\rho^B(a)\cdot q(a)}{\sum_{b\in B}\rho^B(b)\cdot q(b)}$$

with the convention that $p^B(a)=0$ *if the denominator assumes value zero.*

Back

Intuition for the Upper Bound

- Design an automaton that maps each item $a \in A$ to a unique probability ϵ_a of choosing this item during a single investigation

Back

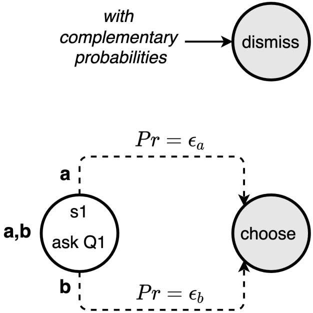
Intuition for the Upper Bound

- Design an automaton that maps each item $a \in A$ to a unique probability ϵ_a of choosing this item during a single investigation
- Show by induction that $f(k) = k - 1$ states are sufficient

Back

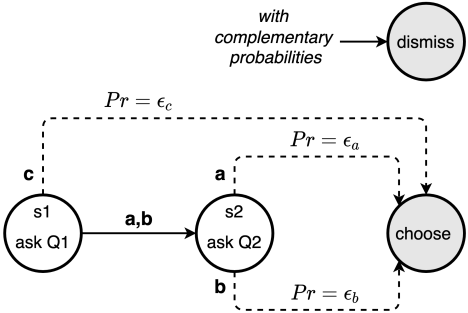
Intuition for the Upper Bound

- Design an automaton that maps each item $a \in A$ to a unique probability ϵ_a of choosing this item during a single investigation
- Show by induction that $f(k) = k - 1$ states are sufficient
- $f(2) = 1$



Intuition for the Upper Bound

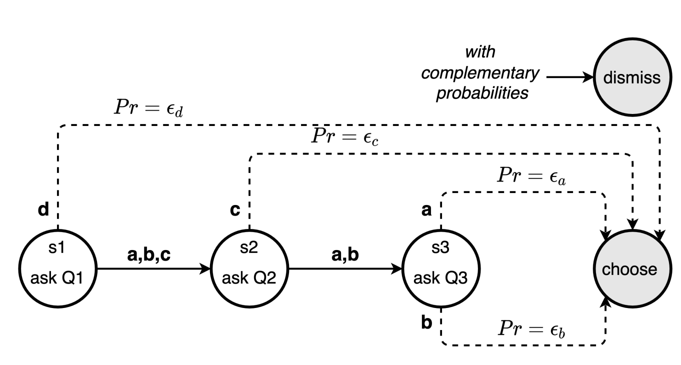
- Design an automaton that maps each item $a \in A$ to a unique probability ϵ_a of choosing this item during a single investigation
- Show by induction that $f(k) = k - 1$ states are sufficient
- $f(k+1) = 1 + f(k) = 1 + k - 1 = k$



Back

Intuition for the Upper Bound

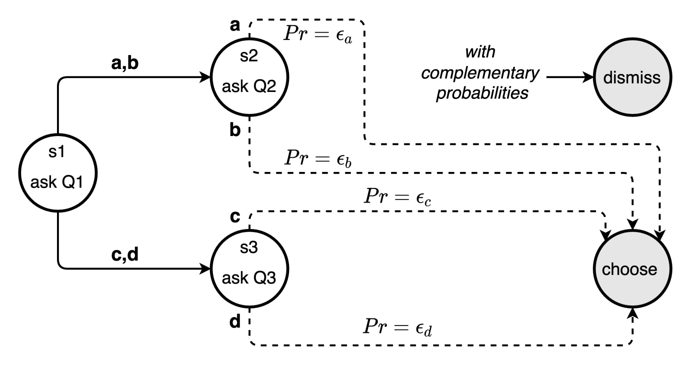
- Design an automaton that maps each item $a \in A$ to a unique probability ϵ_a of choosing this item during a single investigation
- Show by induction that $f(k) = k - 1$ states are sufficient
- $f(k+1) = 1 + f(k) = 1 + k - 1 = k$



Back

Intuition for the Upper Bound

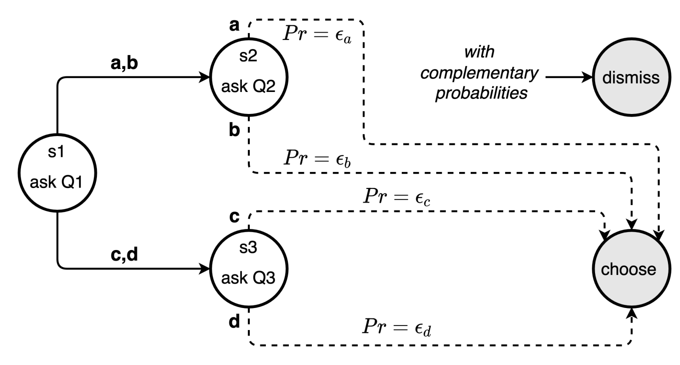
- Design an automaton that maps each item $a \in A$ to a unique probability ϵ_a of choosing this item during a single investigation
- Show by induction that $f(k) = k - 1$ states are sufficient
- $f(k_1 + k_2) = 1 + f(k_1) + f(k_2) = k_1 + k_2 - 1$



Back

Intuition for the Upper Bound

- Design an automaton that maps each item $a \in A$ to a unique probability ϵ_a of choosing this item during a single investigation
- Show by induction that $f(k) = k - 1$ states are sufficient
- Pick sequences $\{\epsilon_a\}_{r=1,2,..}$ for $a \in A$ that solve the choice problem



Back

Existence of Adapted Languages

- WLOG, $\succeq$ is strict:
- Adapted language for k items:

$$\begin{aligned} (i) \quad & a \succ b \quad \Longrightarrow \quad \sum_{i \in N} \lambda_i a_i > \sum_{i \in N} \lambda_i b_i \\ (ii) \quad & \left|\{i \in N | \lambda_i \neq 0\}\right| \; = \; \lceil \log_2 k \rceil \end{aligned}$$

Proof 1:

- Augment the set of items to make $|A| = 2^n$, where $n = \lceil \log_2 k \rceil$
- Consider some collection $\lambda_i > 0$, $i \in \{1, ..., n\}$
- Utility $u(a) = \sum_i \lambda_i a_i$ induces a (strict) preference on vectors of attributes
- Label items in set A accordingly, get an adapted language

Back

Existence of Adapted Languages

- WLOG, $\succeq$ is strict:
- Adapted language for k items:

$$(i) \quad a \succ b \implies \sum_{i \in N} \lambda_i a_i > \sum_{i \in N} \lambda_i b_i$$

$$(ii) \quad \big|\{i \in N | \lambda_i \neq 0\}\big| = \lceil \log_2 k \rceil$$

Proof 2:

- Example: consider $a \succ b \succ c \succ d \succ e \succ f \succ g \succ h$
- Language $Q = \{Q_1, Q_2, Q_3\}$
  - ✓ Q_1 : $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}, e, f, g, h$
  - ✓ Q_2 : $\boldsymbol{a}, \boldsymbol{b}, c, d, \boldsymbol{e}, \boldsymbol{f}, g, h$
  - ✓ Q_3 : $\boldsymbol{a}, b, \boldsymbol{c}, d, \boldsymbol{e}, f, \boldsymbol{g}, h$
- Linear utility: $u(x) = 2^2 \cdot x_1 + 2^1 \cdot x_2 + 2^0 \cdot x_3 = 4x_1 + 2x_2 + x_3$

Back

Lower Bound Characterization Theorem-proof idea for (ii.a)

Theorem (Simplest Languages). *Let* $\succeq$ *have* m *indifference classes, then:*

(i) If Q *is adapted for* $\succeq$*, then* $\kappa_{\succeq}(Q) = 3\lceil \log_2 m \rceil$*;*

(ii) If $(3/4) \cdot 2^n < m \leq 2^n$ *for a natural* n*, then:*

(a) $\kappa_{\succeq}(Q) = 3\lceil \log_2 m \rceil$ *if and only if* Q *is adapted for* $\succeq$*;*

(b) If ψ *solves* $(Q, \succeq)$*, and* $\kappa(\psi) = 3\lceil \log_2 m \rceil$*, then* $\psi \in \Psi^{+}_{\lceil \log_2 m \rceil}$*.*

Recall **Proposition:** *Let* $\succeq$ *have* m *indifference classes, then* Q *is adapted for* $\succeq$ *if and only if there exists* $\psi \in \Psi^{+}_{\lceil \log_2 m \rceil}$ *that solves* $(Q, \succeq)$*.*

Want to prove that when $(3/4) \cdot 2^n < k \leq 2^n$, if ψ solves the choice problem and $\kappa(\psi) \leq 3\lceil \log_2 m \rceil$, then $\psi \in \Psi^{+}_{3\lceil \log_2 m \rceil}$

▸ Back to the Theorem

Lower Bound Characterization Theorem: Proof Sketch (1)

- For each item a, consider a highest-probability path from $s = 1$ to $s = choose$
- Say that $(s, v, j) \in \mathcal{T}$ is a weak link, if $\lim \tau_r(s, v, j) \longrightarrow 0$, otherwise it is a strong link

Lemma. *If the decision rule solves the choice problem, then highest-probability paths for different alternatives use different sets of weak links.*

Lemma. *If ψ solves choice problem with m items, and $\kappa(\psi) = 3\lceil \log_2 k \rceil$, then ψ has n states, $2n$ strong, and n weak links, where $n = \lceil \log_2 k \rceil$.*

Back to the Theorem

Characterization Theorem: Sketch of the Proof (2)

▷ A simple path contains at most 1 link outgoing from a given state

Lemma. *Let the total number of items be* k*,* $n = \lceil \log_2 k \rceil$*, and* $k > (3/4) \cdot 2^n$*. If* ψ *solves the choice problem and* $\kappa(\psi) = 3n$*, then for each pair of weak links there is a highest-probability path that use both these links.*

Corollary. *Let the total number of items be* k*,* $n = \lceil \log_2 k \rceil$*, and* $k > (3/4) \cdot 2^n$*. If* ψ *solves the choice problem and* $\kappa(\psi) = 3n$*, then in every state,* ψ *has exactly one outgoing weak link and exactly two outgoing strong links.*

▸ Back to the Theorem

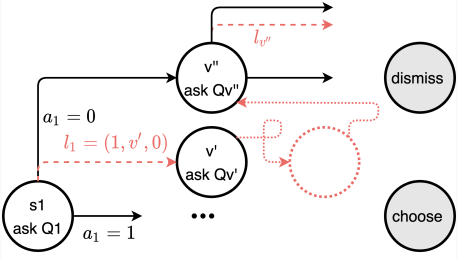
Characterization Theorem: Sketch of the Proof (3)

- WLOG attribute $s \in \{1, ..., n\}$ is investigated in state s.
- WLOG, for each state s:
  - ✓ $\tau(s, v, 1) = 1$ for some v
  - ✓ $\tau(s, v', 0) = \epsilon_s$ and $\tau(s, v'', 0) = 1 - \epsilon_s$ for some v', v'', and $\epsilon_s \longrightarrow 0$
- Recall: to show that $\psi \in \Psi_n^+$, we need to show additionally that there is a labeling of the states such that in the formula above:
  - ✓ $v = v' = s + 1$, where state $n + 1$ denotes *choose*
  - ✓ $v'' \in \{1, .., s\} \cup \{\mathit{dismiss}\}$
- Idea: use induction in n, where $n = \lceil \log_2 k \rceil$, k is the number of items, and condition $k > (3/4) \cdot 2^n$ holds
  - ✓ Induction base: $n = 1$, straightforward
  - ✓ Induction step?

Back to the Theorem

Characterization Theorem: Sketch of the Proof (4)

- Consider $s = 1$, have $\tau(1, v, 1) = 1$, $\tau(1, v', 0) = \epsilon_1$, $\tau(1, v'', 0) = 1 - \epsilon_1$
- $v \notin \{1, \textit{choose}, \textit{dismiss}\}$, since more than 1 item has $a_1 = 1$
- $v' \notin \{1, \textit{choose}, \textit{dismiss}\}$, $v'' \neq \textit{choose}$; otherwise, no more than $2^{n-1} + 1 \leq (3/4) \cdot 2^n$ different subsets of weak links used
- Towards a contradiction, assume $v'' \notin \{1, \textit{dismiss}\}$



- Highest-probability path cannot include both weak links l_1 and $l_{v''}$, in contradiction

Back to the Theorem

Characterization Theorem: Sketch of the Proof (5)

- We know: $\tau(1, v, 1) = 1$, $\tau(1, v', 0) = \epsilon_1$, $\tau(1, v'', 0) = 1 - \epsilon_1$
  - ✓ $v, v' \notin \{1, \mathit{choose}, \mathit{dismiss}\}$, $v'' \in \{1, \mathit{dismiss}\}$
- At least one of the two statements should hold:
  - ✓ $\big|\{a \in A | a_i = 1\}\big| > (3/4) \cdot 2^{n-1}$
  - ✓ $\big|\{a \in A | a_i = 0\}\big| > (3/4) \cdot 2^{n-1}$
- Let $\big|\{a \in A | a_i = 1\}\big| > (3/4) \cdot 2^{n-1}$, consider rule ψ':
  - ✓ Delete state $s = 1$ in rule ψ and its outgoing links
  - ✓ Redirect each link that ends at $s = 1$ in ψ to $s = \mathit{dismiss}$ in ψ'
  - ✓ Make state v the first state in ψ'
- ψ' solves the problem constrained to items $\{a \in A | a_i = 1\}$
  - ✓ $\kappa(\psi') \leq 3n - 3$
  - ✓ Use induction assumption to find configuration of links outgoing from all other states except of $s = 1$ in ψ

▸ Back to the Theorem

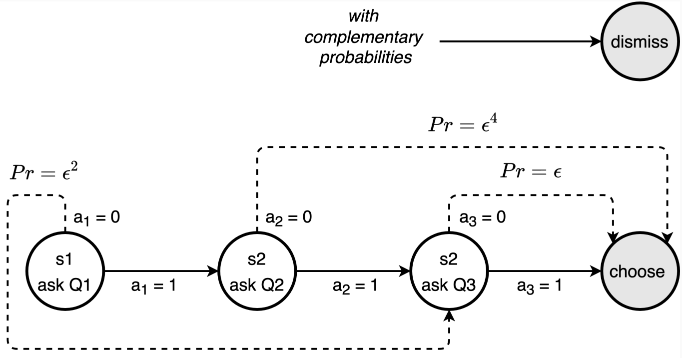
Characterization Theorem: Sketch of the Proof (6)

- Last statement to prove: that $v' = v$.
- Assume $v' \neq v$, then weak link $(1, v', 0)$ and weak link l_v, outgoing from state v, cannot be in the same highest-probability path, contradiction
- Similar arguments work if $\left|\{a \in A | a_i = 0\}\right| > (3/4) \cdot 2^{n-1}$
  - ✓ Note that $\left|\{a \in A | a_i = 0\}\right| \leq (1/2) \cdot 2^n$
  - ✓ Hence $\left|\{a \in A | a_i = 0\}\right| > (1/4) \cdot 2^n$
  - ✓ If $v \neq v'$, a weak link outgoing from v' is not used in any highest-probability paths for items with $a_1 = 1$
  - ✓ Thus, no more than $(1/4) \cdot 2^n$ sets of weak links used in highest-probability paths for items with $a_1 = 1$, contradiction

Back to the Theorem

(Counter) Example

- $k = 5$, so $n = \lceil \log_2 5 \rceil = 3$, $k = 5 \leq (3/4) \cdot 2^3 = 6$
- 111 $\succ$ 110 $\succ$ 011 $\succ$ 000 $\succ$ 100



(Counter) Example 2

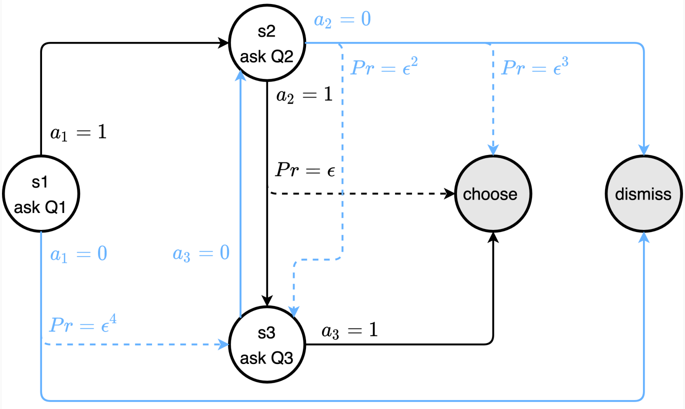
- $111 \succ 110 \succ 101 \succ 100 \succ 001 \succ 010 \succ 000$

(Counter) Example 2

- $111 \succ 110 \succ 101 \succ 100 \succ 001 \succ 010 \succ 000$

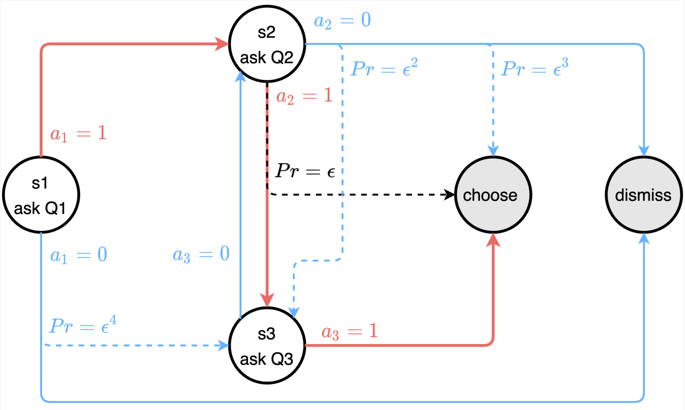
(Counter) Example 2

- $111 \succ 110 \succ 101 \succ 100 \succ 001 \succ 010 \succ 000$



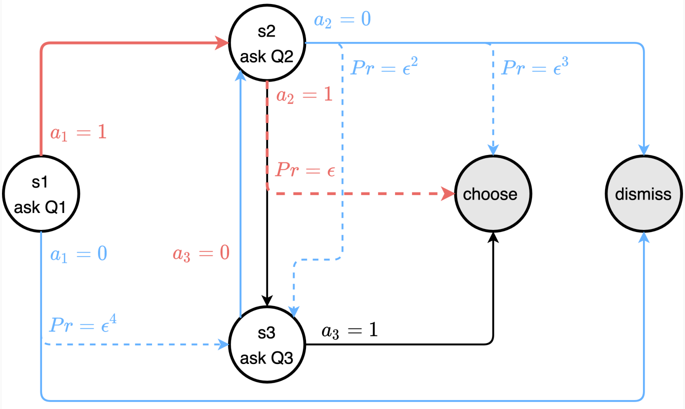
(Counter) Example 2

- ▷ $\color{red}{111} \succ 110 \succ 101 \succ 100 \succ 001 \succ 010 \succ 000$



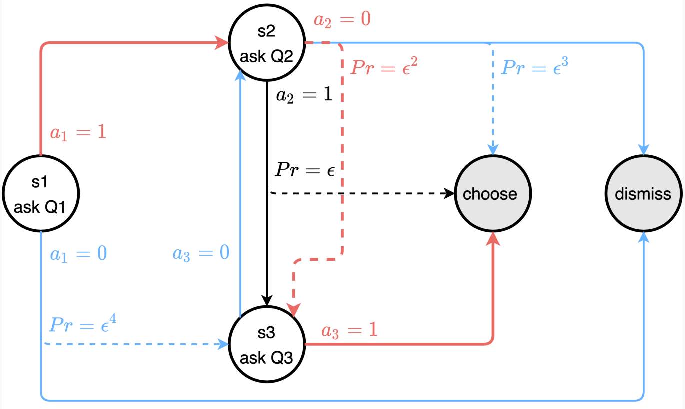
(Counter) Example 2

▷ $111 \succ$ 110 $\succ 101 \succ 100 \succ 001 \succ 010 \succ 000$



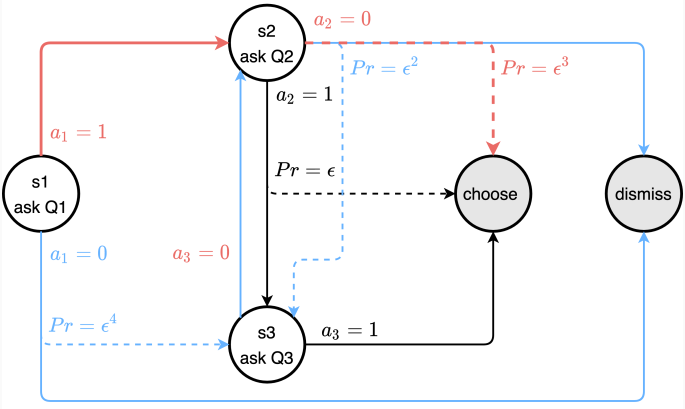
(Counter) Example 2

- $111 \succ 110 \succ 101 \succ 100 \succ 001 \succ 010 \succ 000$



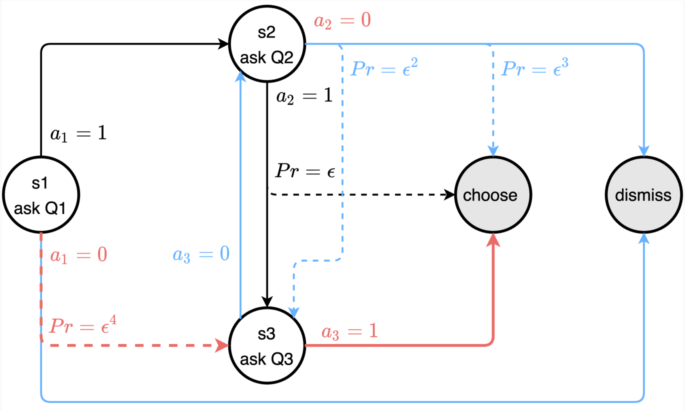
(Counter) Example 2

- $111 \succ 110 \succ 101 \succ 100 \succ 001 \succ 010 \succ 000$



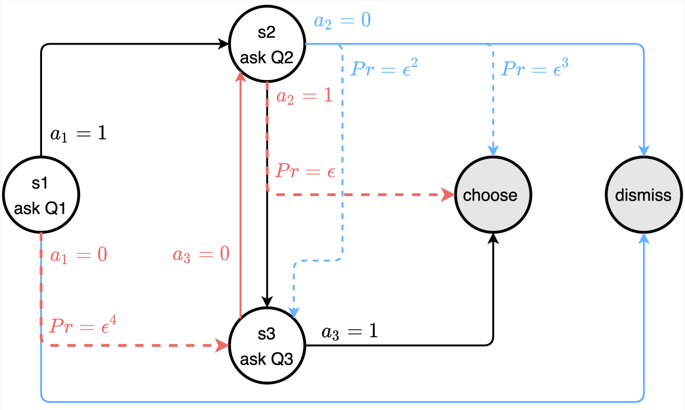
(Counter) Example 2

▷ $111 \succ 110 \succ 101 \succ 100 \succ$ 001 $\succ 010 \succ 000$



(Counter) Example 2

- ▷ 111 ≻ 110 ≻ 101 ≻ 100 ≻ 001 ≻ 010 ≻ 000



(Counter) Example 2

- ▷ $111 \succ 110 \succ 101 \succ 100 \succ 001 \succ 010 \succ 000$