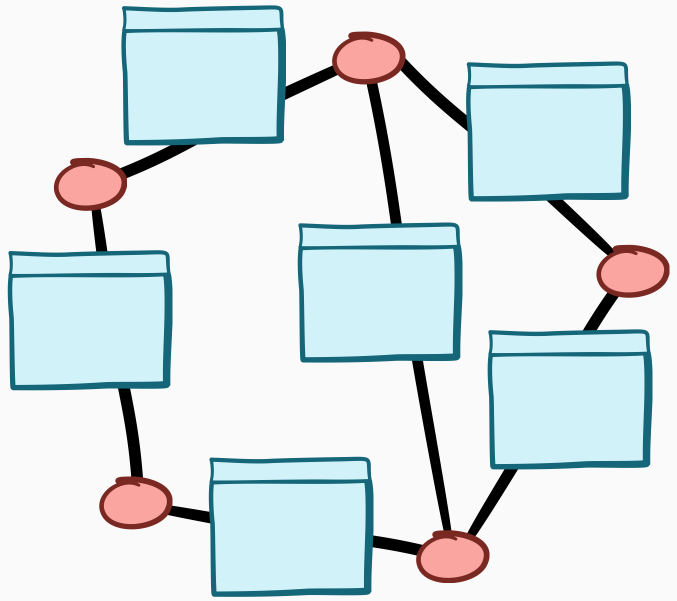
Local Hedging *approximately solves* Pandora's Box Problems *with* Nonobligatory Inspection

Ziv Scully
Cornell ORIE

Laura Doval
Columbia Business School



In uncertain environments, when is the

value of new information

worth the cost of obtaining it?

In uncertain environments, when is the

value of new information

worth the cost of obtaining it?

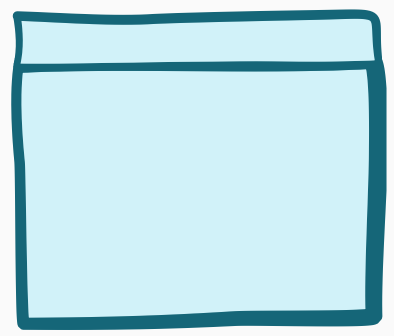
In stochastic control, when can we

decompose hard problems

into more tractable subproblems?

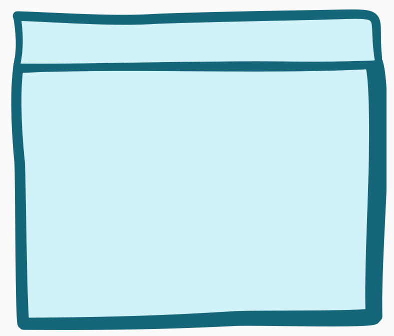
Modeling cost of information

Pandora's box model:



Modeling cost of information

Pandora's box model:

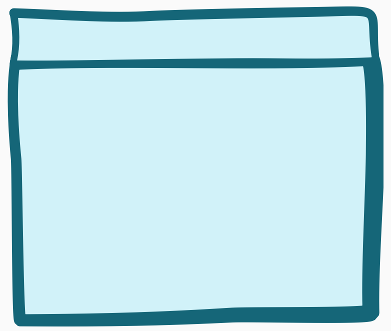


- Opening cost c

Modeling cost of information

Pandora's box model:

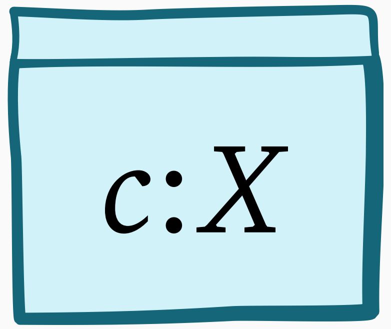
- Opening cost c
- Hidden price X



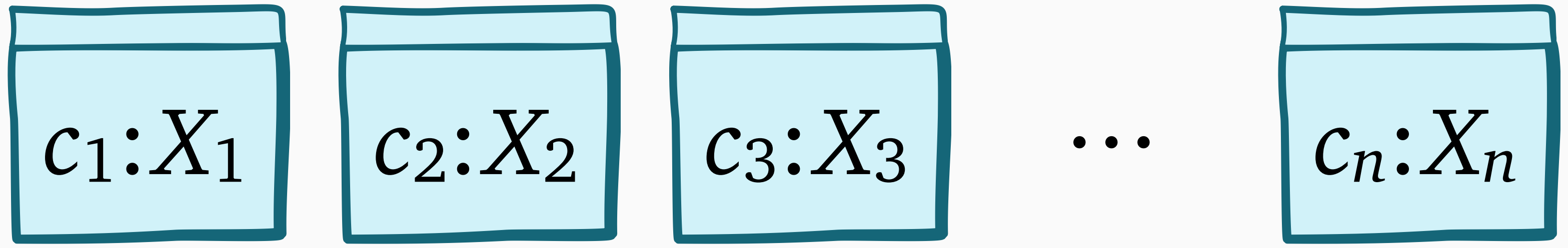
Modeling cost of information

Pandora's box model:

- Opening cost c
- Hidden price X

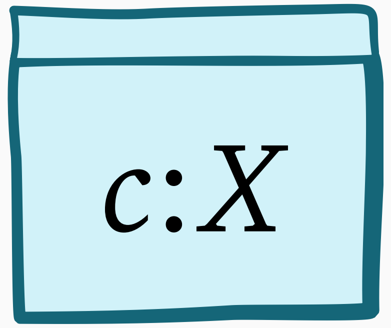


Modeling cost of information

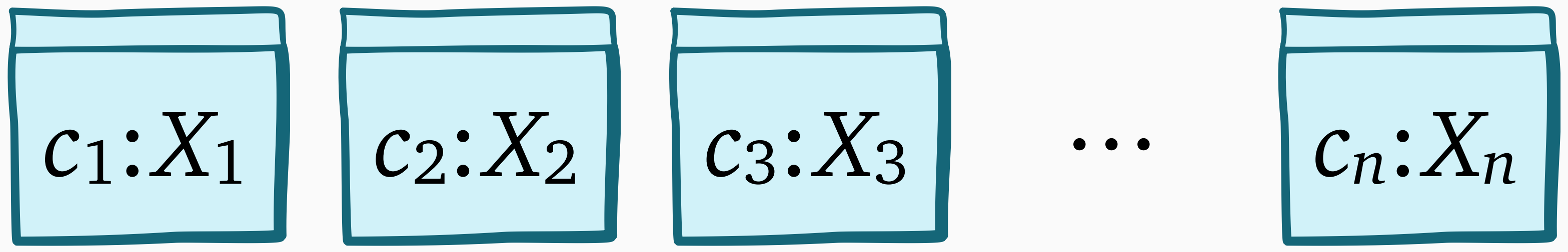


Pandora's box model:

- Opening cost c
- Hidden price X

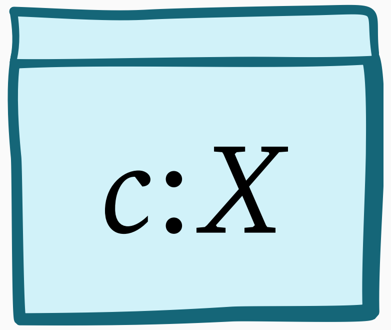


Modeling cost of information



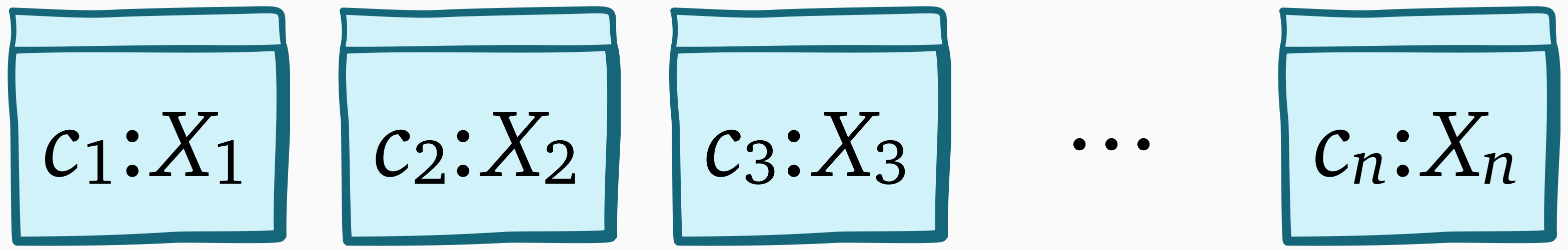
Pandora's box model:

- Opening cost c
- Hidden price X



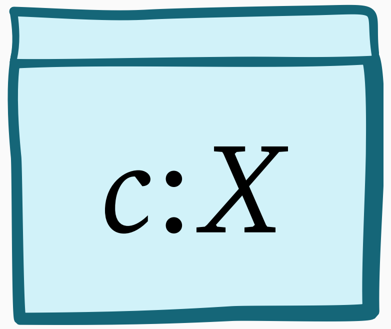
Classic problem:

Modeling cost of information



Pandora's box model:

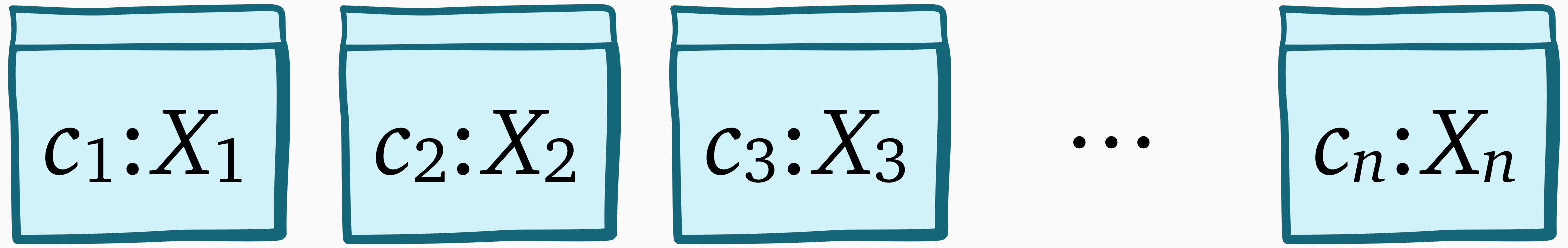
- Opening cost c
- Hidden price X



Classic problem:

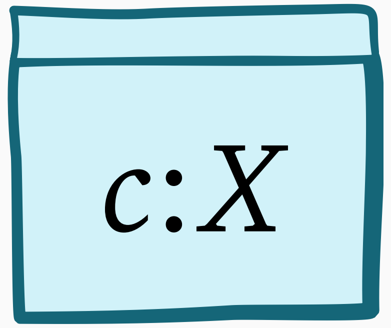
- Open boxes one at a time

Modeling cost of information



Pandora's box model:

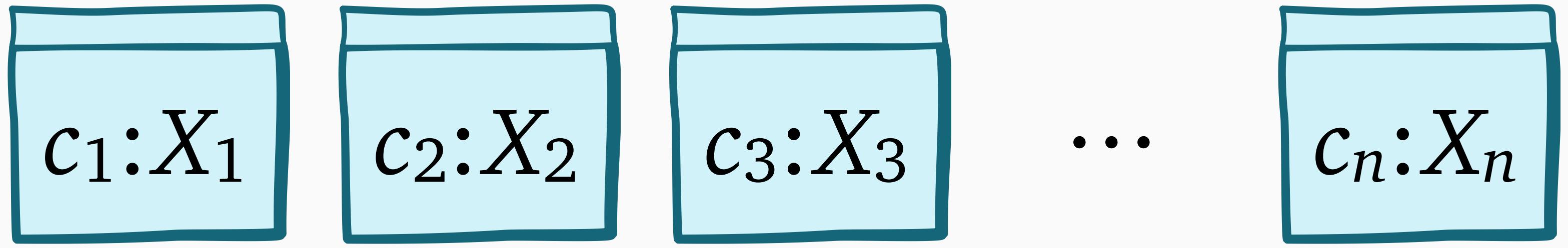
- Opening cost c
- Hidden price X



Classic problem:

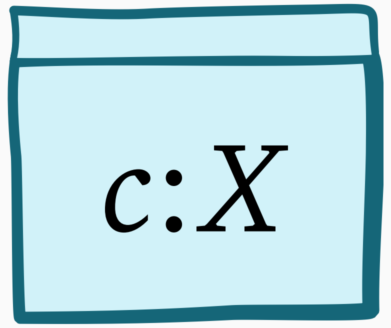
- Open boxes one at a time
- Stop by selecting open box

Modeling cost of information



Pandora's box model:

- Opening cost c
- Hidden price X

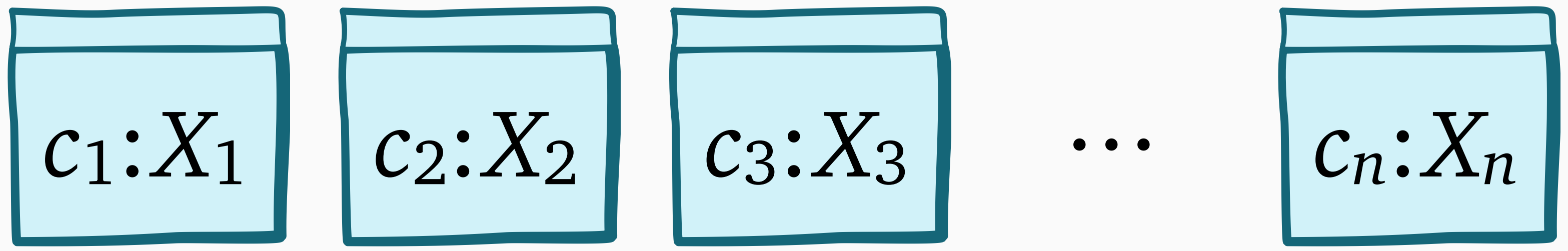


Classic problem:

- Open boxes one at a time
- Stop by selecting open box

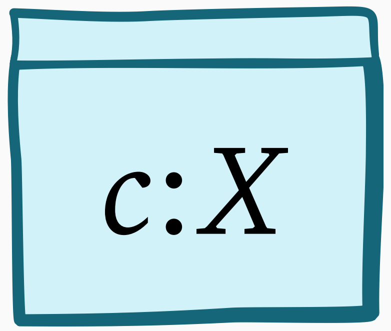
Goal: minimize $\mathbf{E}\left[\sum_{i \text{ opened}} c_i + \sum_{j \text{ selected}} X_j\right]$

Modeling cost of information



Pandora's box model:

- Opening cost c
- Hidden price X



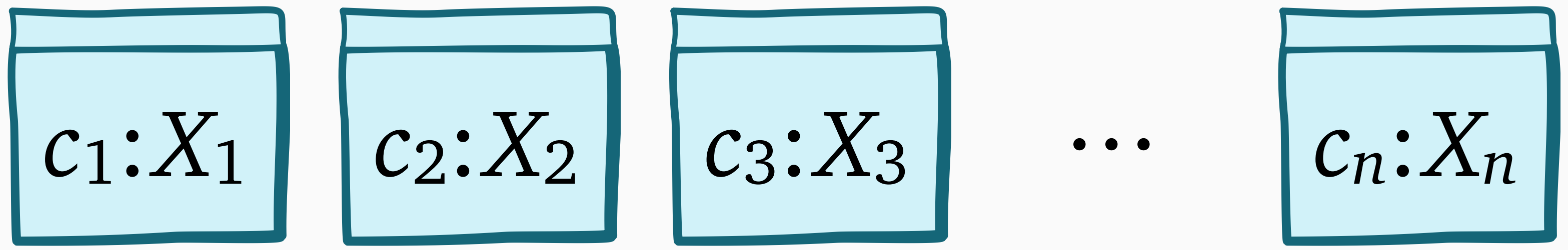
Classic problem:

- Open boxes one at a time
- Stop by selecting open box

Combinatorial problems:
select *admissible set* of open boxes

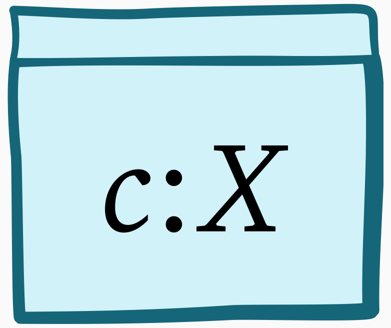
Goal: minimize $\mathbf{E}\left[\sum_{i \text{ opened}} c_i + \sum_{j \text{ selected}} X_j\right]$

Modeling cost of information



Pandora's box model:

- Opening cost c
- Hidden price X

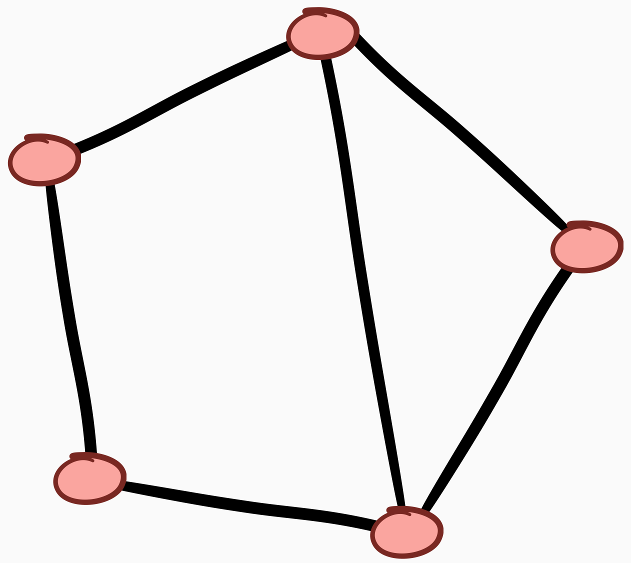


Classic problem:

- Open boxes one at a time
- Stop by selecting open box

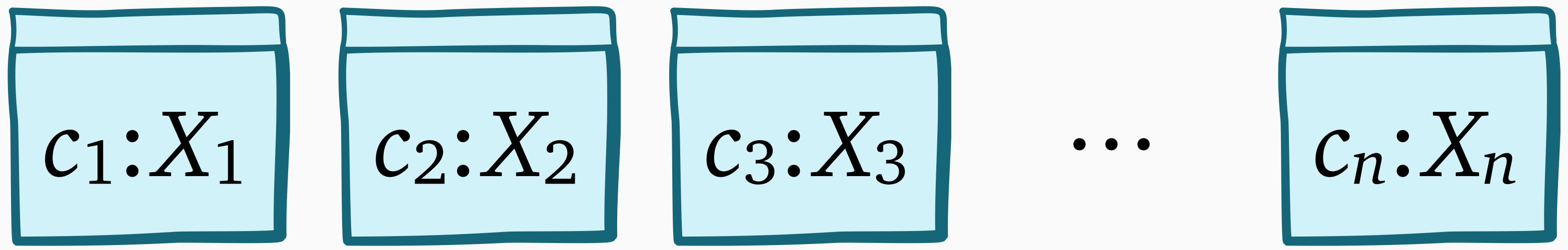
Goal: minimize $\mathbf{E}\left[\sum_{i \text{ opened}} c_i + \sum_{j \text{ selected}} X_j\right]$

Combinatorial problems:
select *admissible set* of open boxes



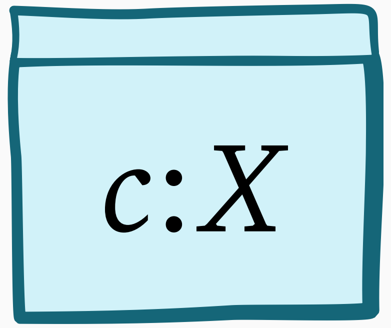
Example: build a spanning tree

Modeling cost of information



Pandora's box model:

- Opening cost c
- Hidden price X

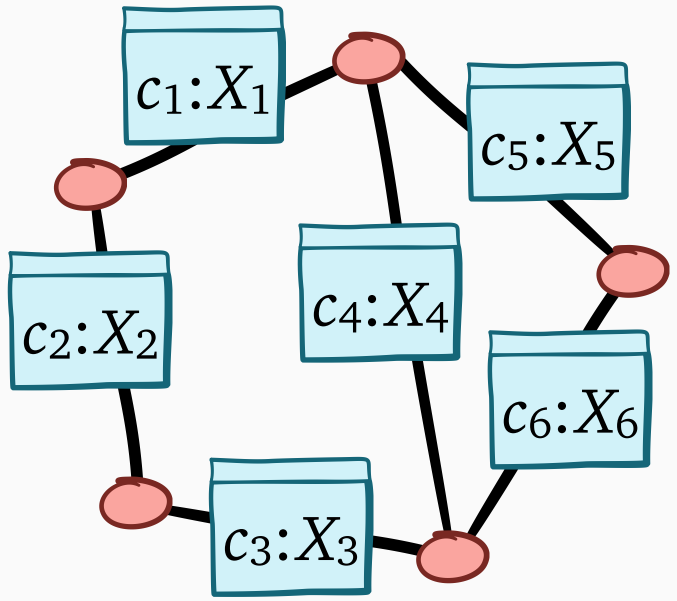


Classic problem:

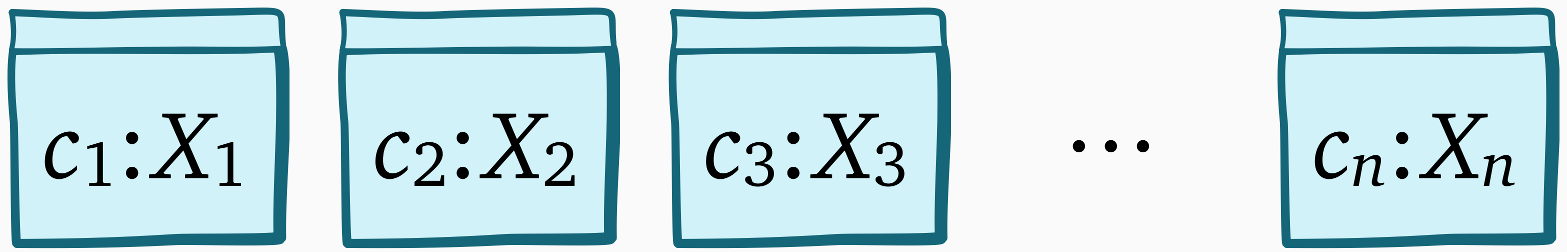
- Open boxes one at a time
- Stop by selecting open box

Goal: minimize $\mathbf{E}\Big[\sum_{i \text{ opened}} c_i + \sum_{j \text{ selected}} X_j\Big]$

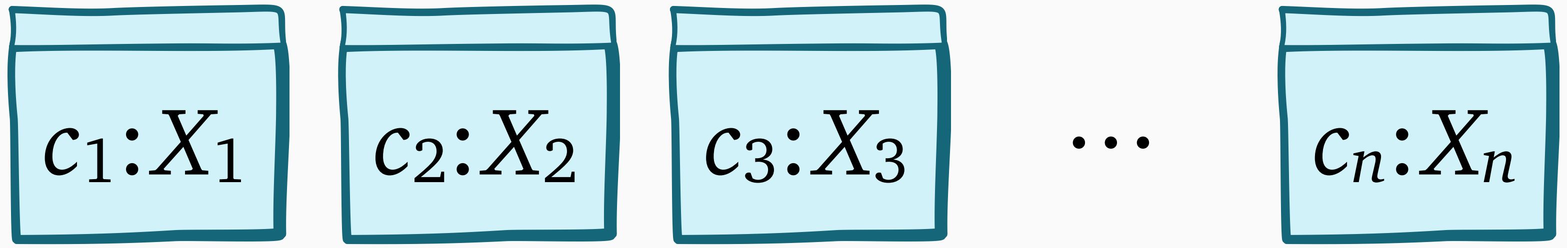
Combinatorial problems:
select *admissible set* of open boxes



Example: build a spanning tree



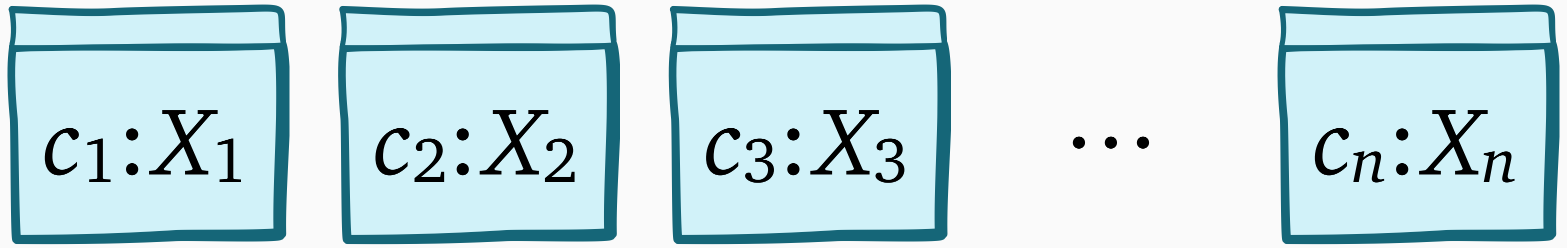
Goal: minimize $\mathbf{E}\left[\sum_{i \text{ opened}} c_i + \sum_{j \text{ selected}} X_j\right]$



Goal: minimize $\mathbf{E}\Big[\sum_{i \text{ opened}} c_i + \sum_{j \text{ selected}} X_j\Big]$

Multifaceted decision

Should we stop? If not, which box should we open?



Goal: minimize $\mathbf{E}\left[\sum_{i \text{ opened}} c_i + \sum_{j \text{ selected}} X_j\right]$

Multifaceted decision

Should we stop? If not, which box should we open?

Large state space

Grows exponentially with number of boxes n

$c_1{:}X_1$ $c_2{:}X_2$ $c_3{:}X_3$ $\cdots$ $c_n{:}X_n$

Goal: minimize $\mathbf{E}\left[\sum_{i \text{ opened}} c_i + \sum_{j \text{ selected}} X_j\right]$

Multifaceted decision

Should we stop? If not, which box should we open?

Large state space

Grows exponentially with number of boxes n

Combinatorial constraints

Can make problem hard even without uncertainty

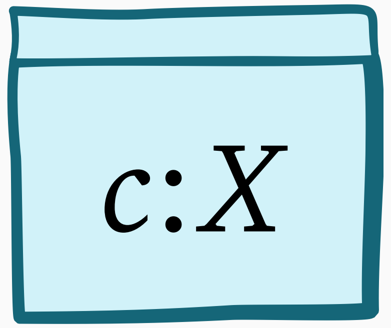
Decomposition for Pandora's box

Step 1: *rate* each box separately

Step 2: *act* on box of best rating

Decomposition for Pandora's box

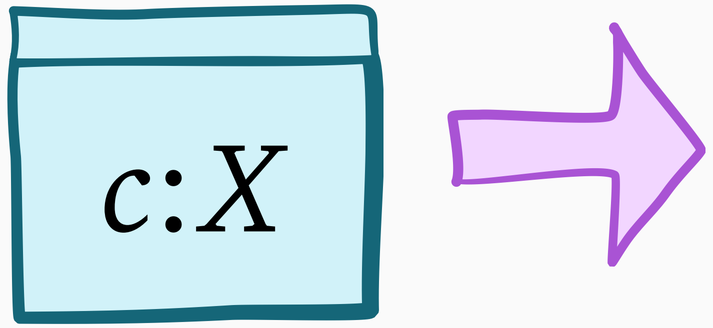
Step 1: *rate* each box separately



Step 2: *act* on box of best rating

Decomposition for Pandora's box

Step 1: *rate* each box separately



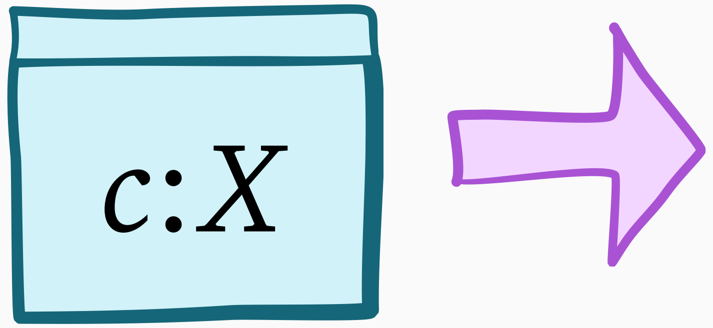
Gittins index:

$g(c{:}X)$

Step 2: *act* on box of best rating

Decomposition for Pandora’s box

Step 1: *rate* each box separately



Gittins index:

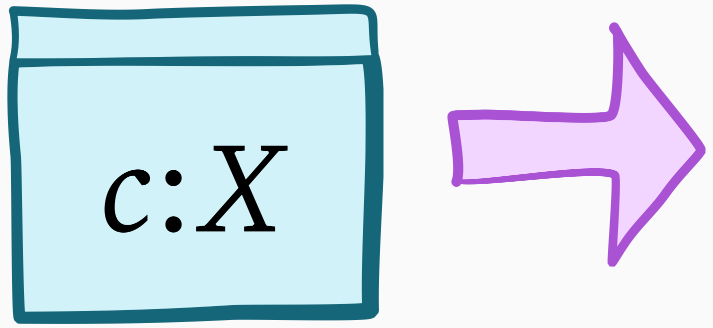
$g(c{:}X)$

lower is better

Step 2: *act* on box of best rating

Decomposition for Pandora's box

Step 1: *rate* each box separately

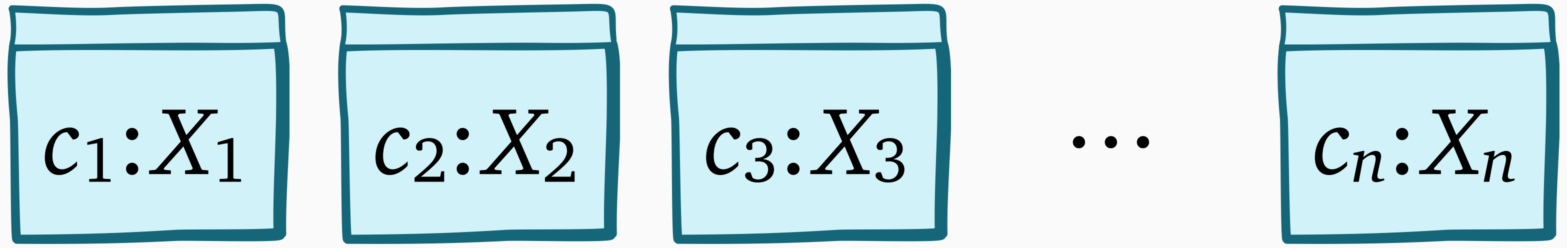


Gittins index:

$g(c\!:\!X)$

lower is better

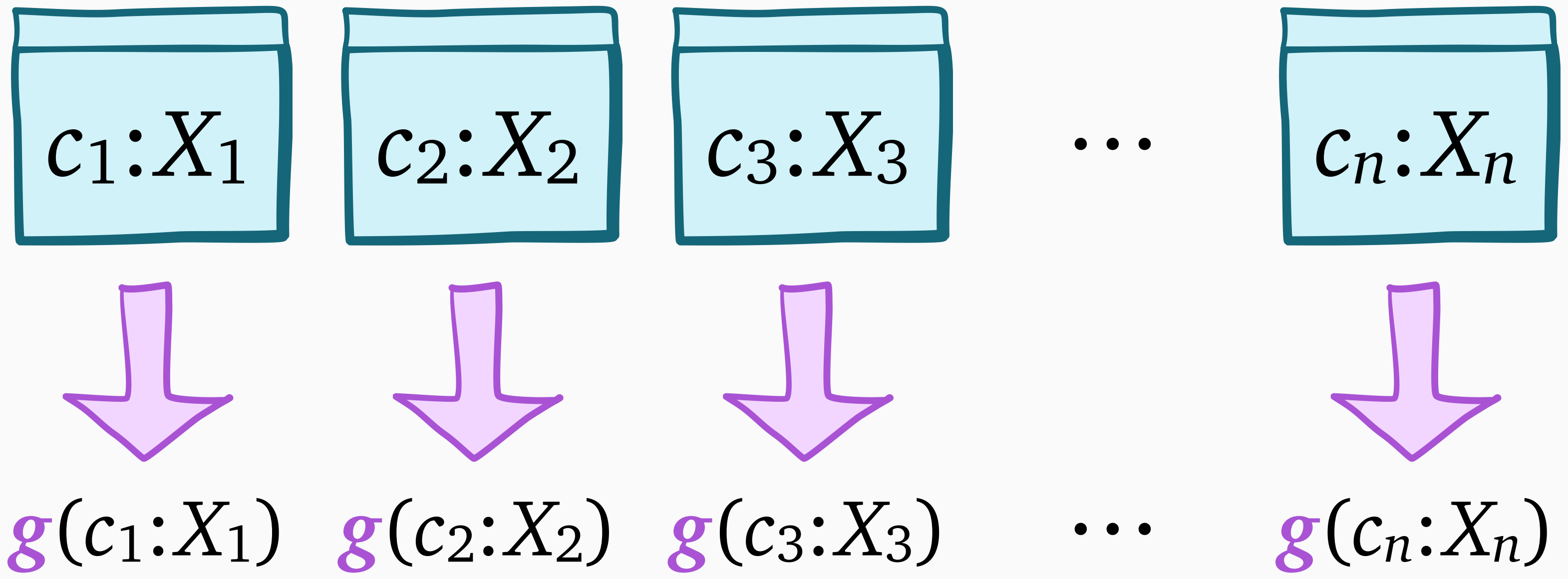
Step 2: *act* on box of best rating



Decomposition for Pandora's box

Step 1: *rate* each box separately

Step 2: *act* on box of best rating



Decomposition for Pandora's box

Step 1: *rate* each box separately

$c{:}X$

Gittins index:

$g(c{:}X)$

lower is better

Step 2: *act* on box of best rating

$c_1{:}X_1$ $c_2{:}X_2$ $c_3{:}X_3$ $\cdots$ $c_n{:}X_n$

$g(c_1{:}X_1)$ $g(c_2{:}X_2)$ $g(c_3{:}X_3)$ $\cdots$ $g(c_n{:}X_n)$

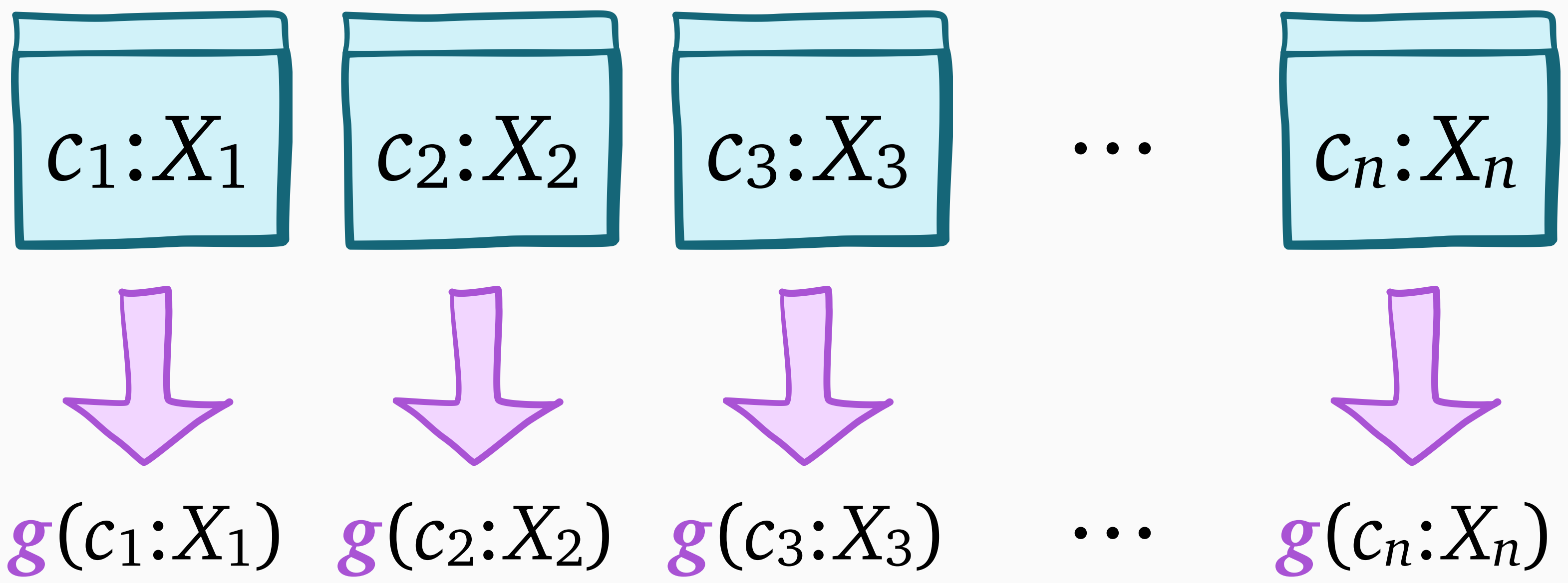
Gittins policy: if box of least Gittins index is…

- *closed:* open it
- *open:* select it

Decomposition for Pandora's box

Step 1: *rate* each box separately

Step 2: *act* on box of best rating



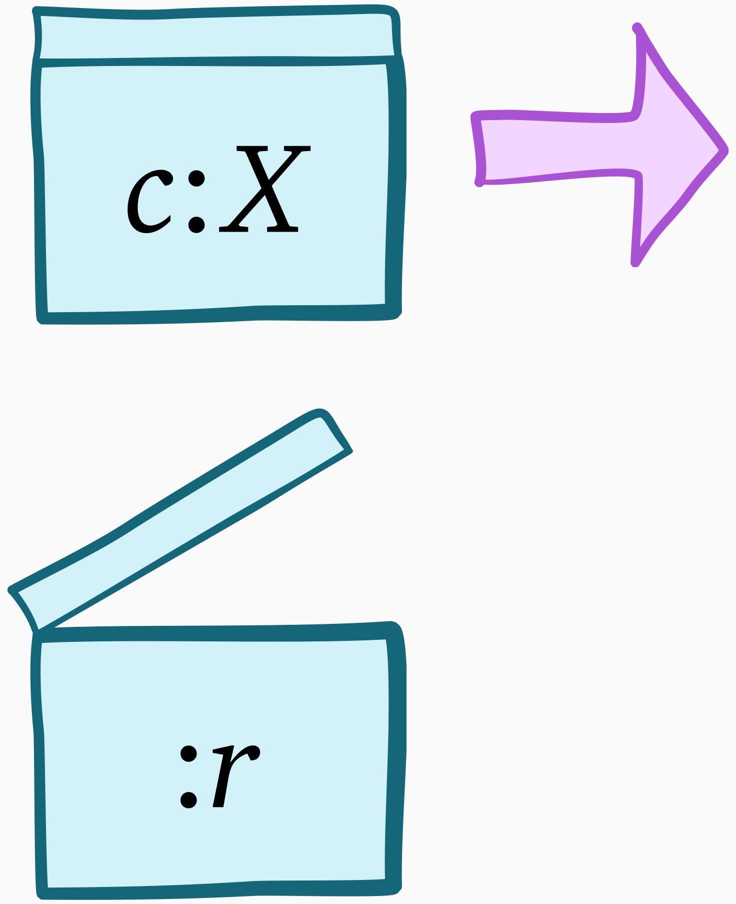
Gittins **policy:** if box of least Gittins index is…

- *closed:* open it
- *open:* select it

} *act* on it

Decomposition for Pandora's box

Step 1: *rate* each box separately

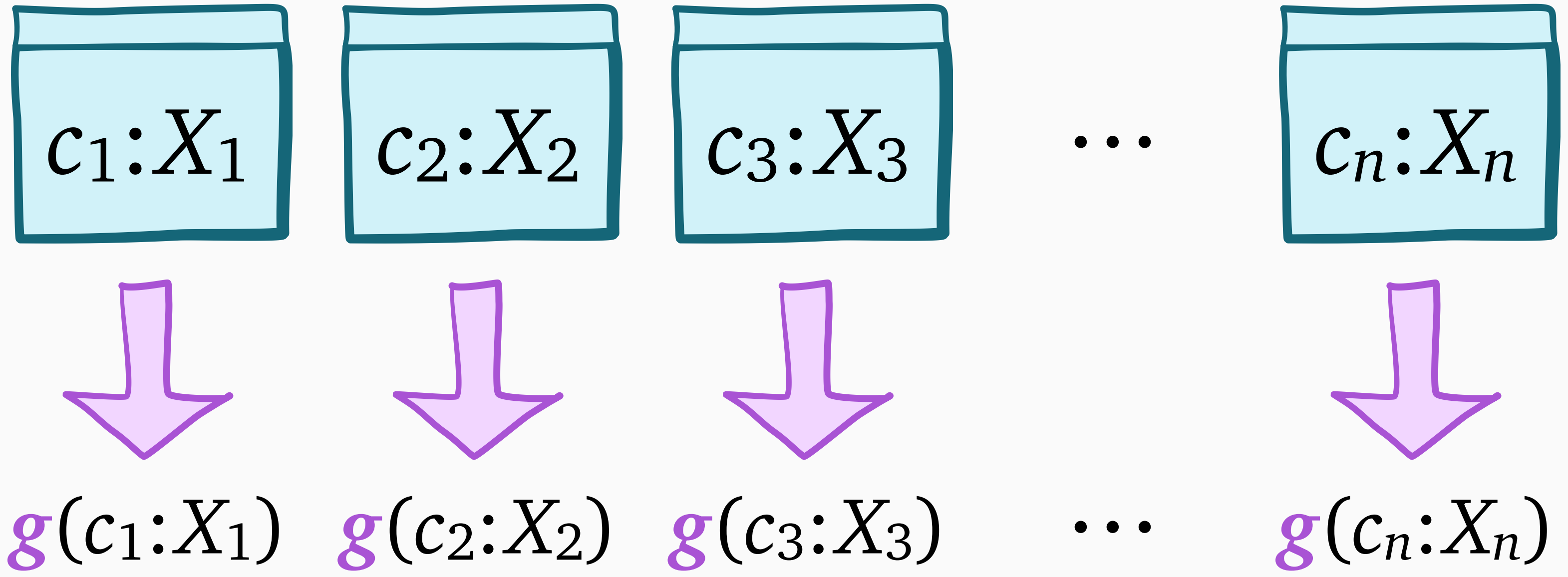


Gittins index:

$g(c{:}X)$

lower is better

Step 2: *act* on box of best rating



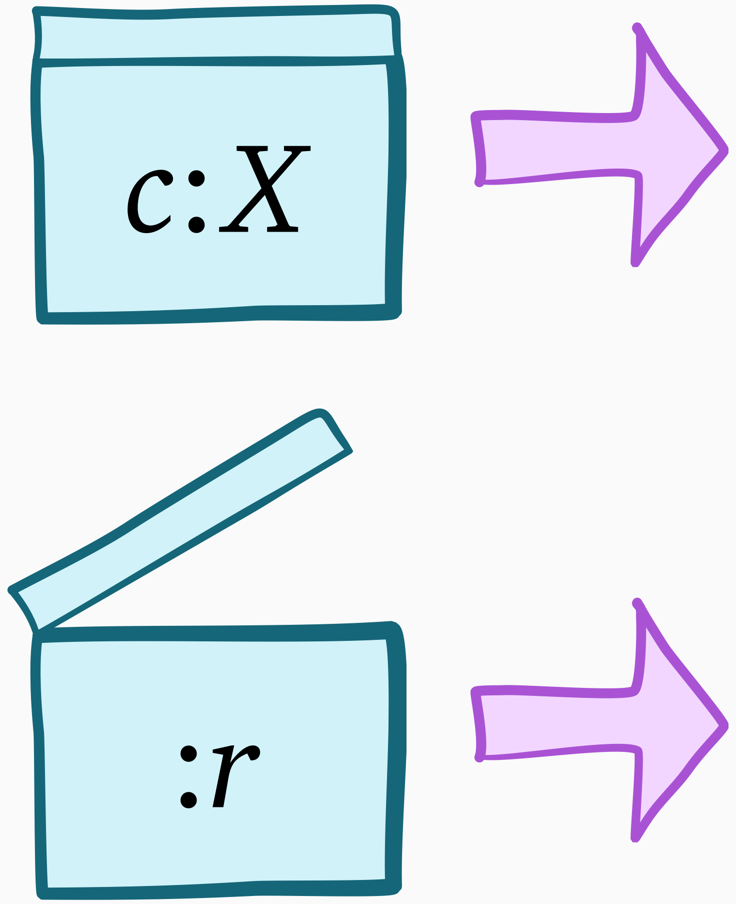
Gittins policy: if box of least **Gittins** index is…

- *closed:* open it
- *open:* select it

} *act* on it

Decomposition for Pandora's box

Step 1: *rate* each box separately



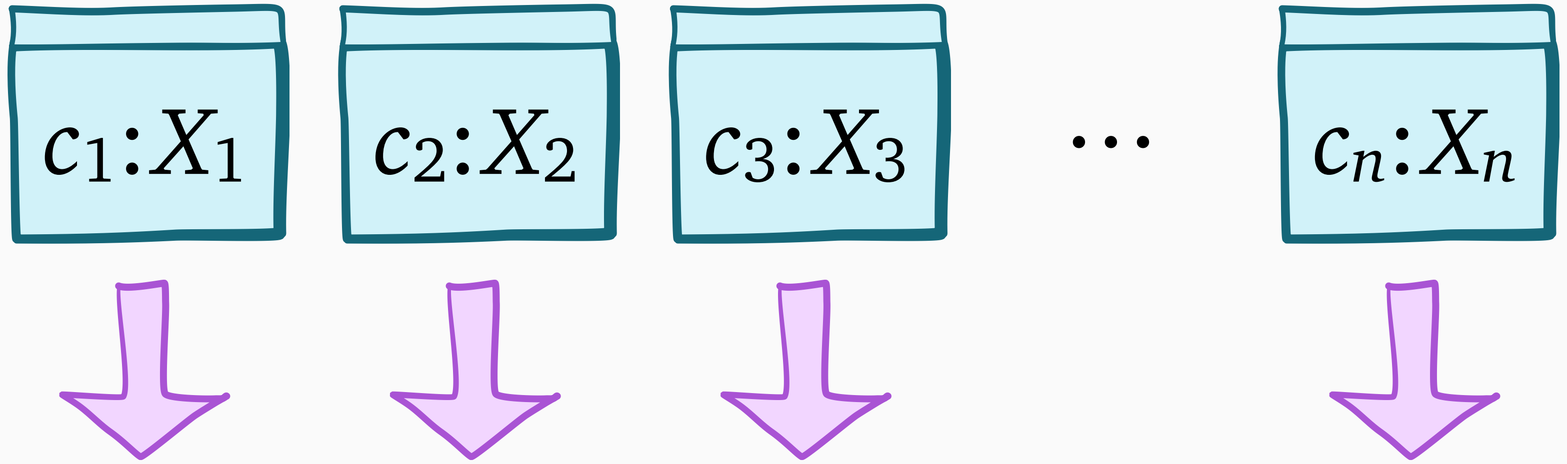
Gittins index:

$g(c{:}X)$

lower is better

$g(:r) = r$

Step 2: *act* on box of best rating



$g(c_1{:}X_1)$ $g(c_2{:}X_2)$ $g(c_3{:}X_3)$ $\cdots$ $g(c_n{:}X_n)$

Gittins policy: if box of least **Gittins** index is…

- *closed:* open it
- *open:* select it

} *act* on it

Decomposition for Pandora's box

Step 1: *rate* each box separately

$c{:}X$ → **Gittins index:** $g(c{:}X)$

lower is better

$:r$ → $g(:r) = r$

Theorem [Weitzman, 1979]: the **Gittins** policy is optimal

Step 2: *act* on box of best rating

$c_1{:}X_1$ $c_2{:}X_2$ $c_3{:}X_3$ $\cdots$ $c_n{:}X_n$

$g(c_1{:}X_1)$ $g(c_2{:}X_2)$ $g(c_3{:}X_3)$ $\cdots$ $g(c_n{:}X_n)$

Gittins policy: if box of least **Gittins** index is…

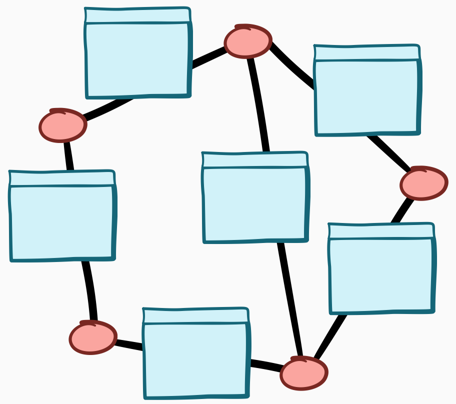
- *closed:* open it
- *open:* select it

} *act* on it

Classic problem

$c_1{:}X_1$ $c_2{:}X_2$ $c_3{:}X_3$ $\cdots$ $c_n{:}X_n$

Combinatorial problems



$c_1{:}X_1$ $c_2{:}X_2$ $c_3{:}X_3$ $\cdots$ $c_n{:}X_n$

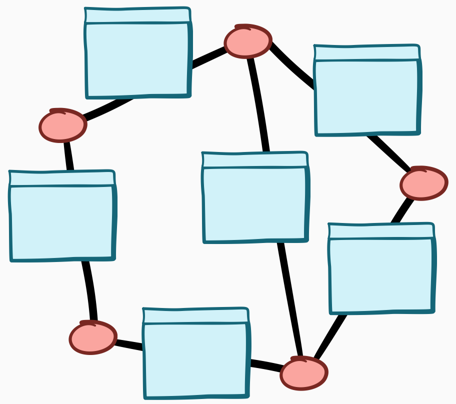
Classic problem

$c_1{:}X_1 \quad c_2{:}X_2 \quad c_3{:}X_3 \quad \cdots \quad c_n{:}X_n$

$\downarrow \quad \downarrow \quad \downarrow \quad \quad \downarrow$

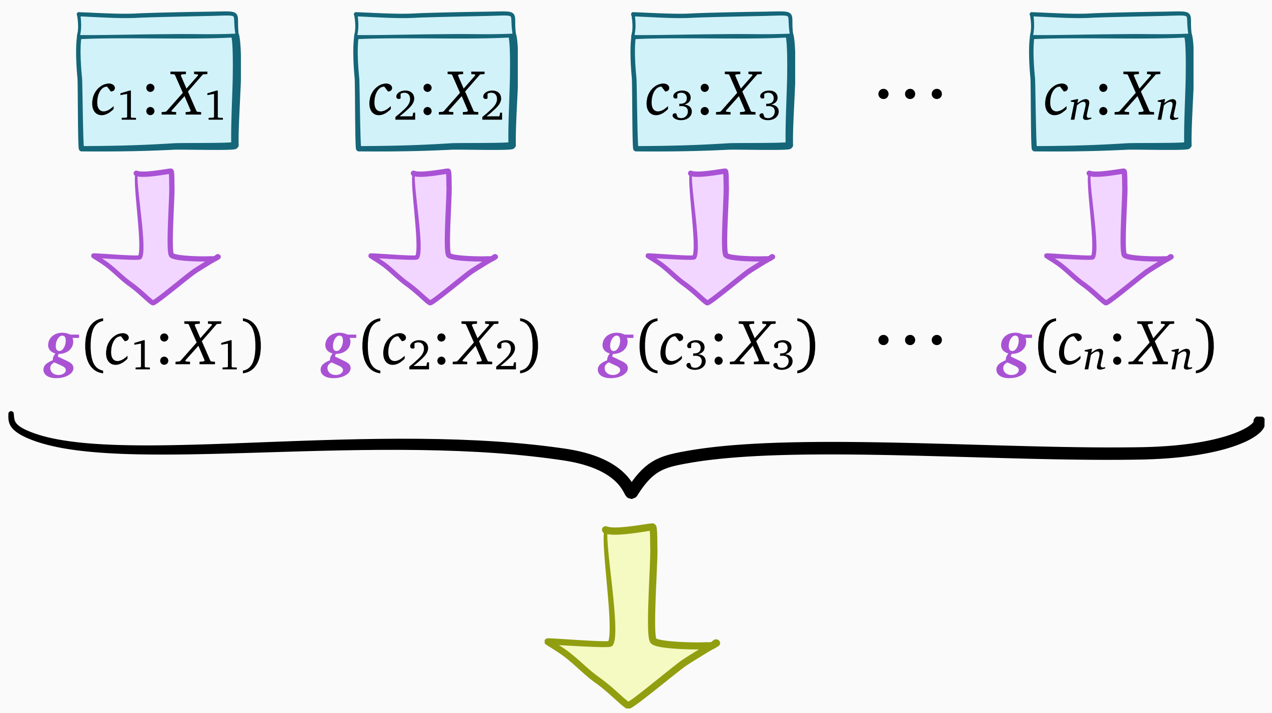
$g(c_1{:}X_1) \quad g(c_2{:}X_2) \quad g(c_3{:}X_3) \quad \cdots \quad g(c_n{:}X_n)$

Combinatorial problems



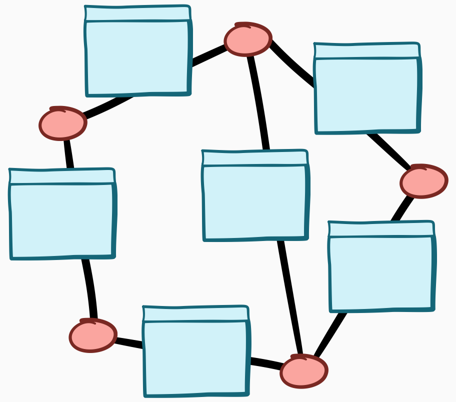
$c_1{:}X_1 \quad c_2{:}X_2 \quad c_3{:}X_3 \quad \cdots \quad c_n{:}X_n$

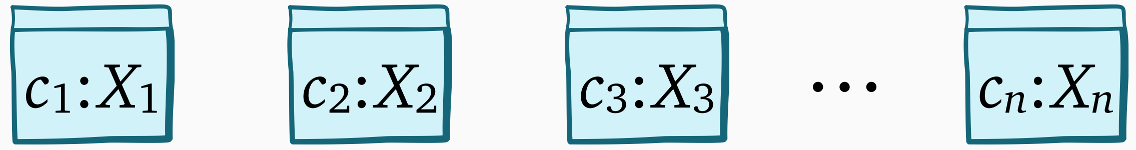
Classic problem



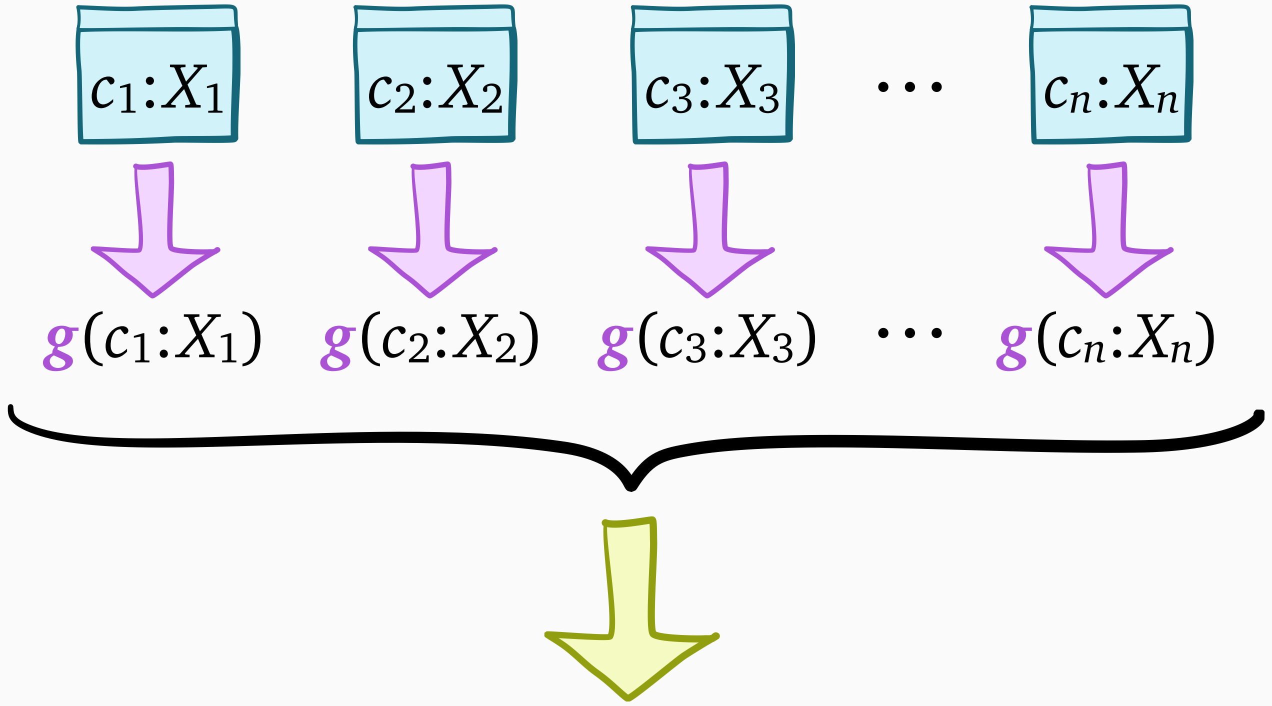
Deterministic algorithm:
pick smallest number

Combinatorial problems



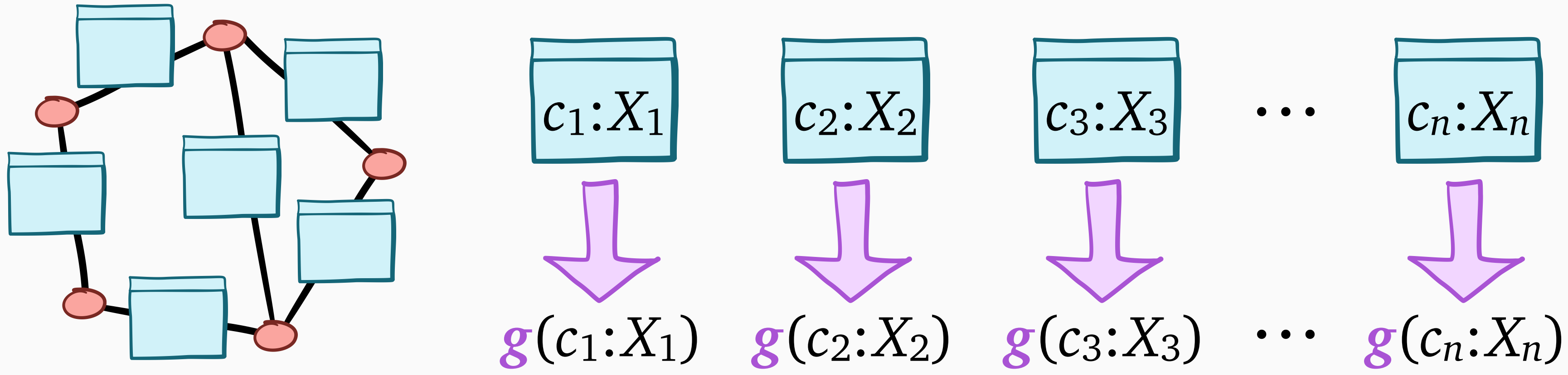


Classic problem



Deterministic algorithm:
pick smallest number

Combinatorial problems



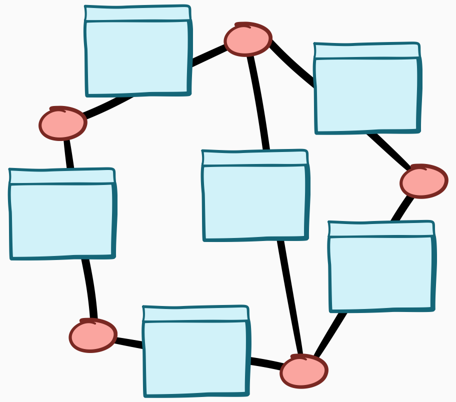
Classic problem

$c_1{:}X_1$ $c_2{:}X_2$ $c_3{:}X_3$ $\cdots$ $c_n{:}X_n$

$g(c_1{:}X_1)$ $g(c_2{:}X_2)$ $g(c_3{:}X_3)$ $\cdots$ $g(c_n{:}X_n)$

Deterministic algorithm:
pick smallest number

Combinatorial problems



$c_1{:}X_1$ $c_2{:}X_2$ $c_3{:}X_3$ $\cdots$ $c_n{:}X_n$

$g(c_1{:}X_1)$ $g(c_2{:}X_2)$ $g(c_3{:}X_3)$ $\cdots$ $g(c_n{:}X_n)$

Deterministic algorithm:
"greedy" algorithm **Alg**

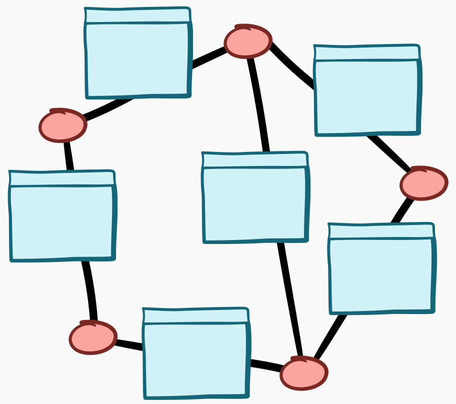
Classic problem

$c_1{:}X_1$ $c_2{:}X_2$ $c_3{:}X_3$ $\cdots$ $c_n{:}X_n$

$g(c_1{:}X_1)$ $g(c_2{:}X_2)$ $g(c_3{:}X_3)$ $\cdots$ $g(c_n{:}X_n)$

Deterministic algorithm:
pick smallest number

Combinatorial problems



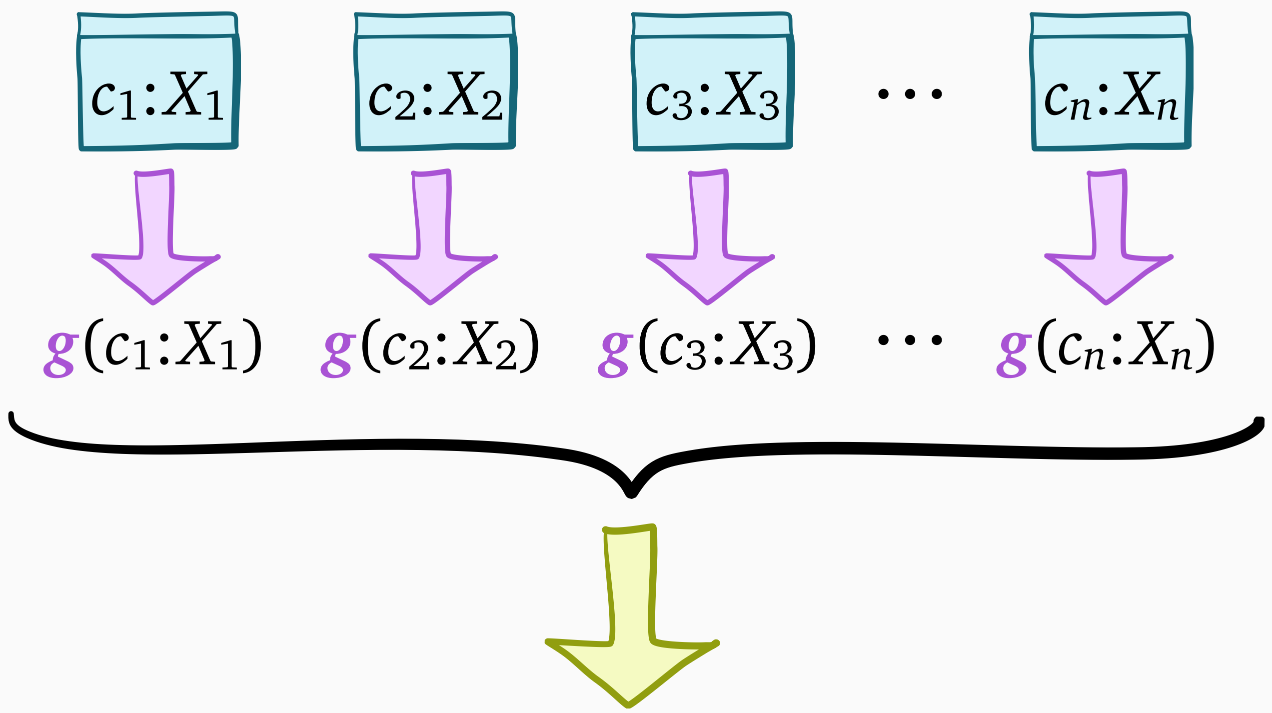
$c_1{:}X_1$ $c_2{:}X_2$ $c_3{:}X_3$ $\cdots$ $c_n{:}X_n$

$g(c_1{:}X_1)$ $g(c_2{:}X_2)$ $g(c_3{:}X_3)$ $\cdots$ $g(c_n{:}X_n)$

Gittins + Alg

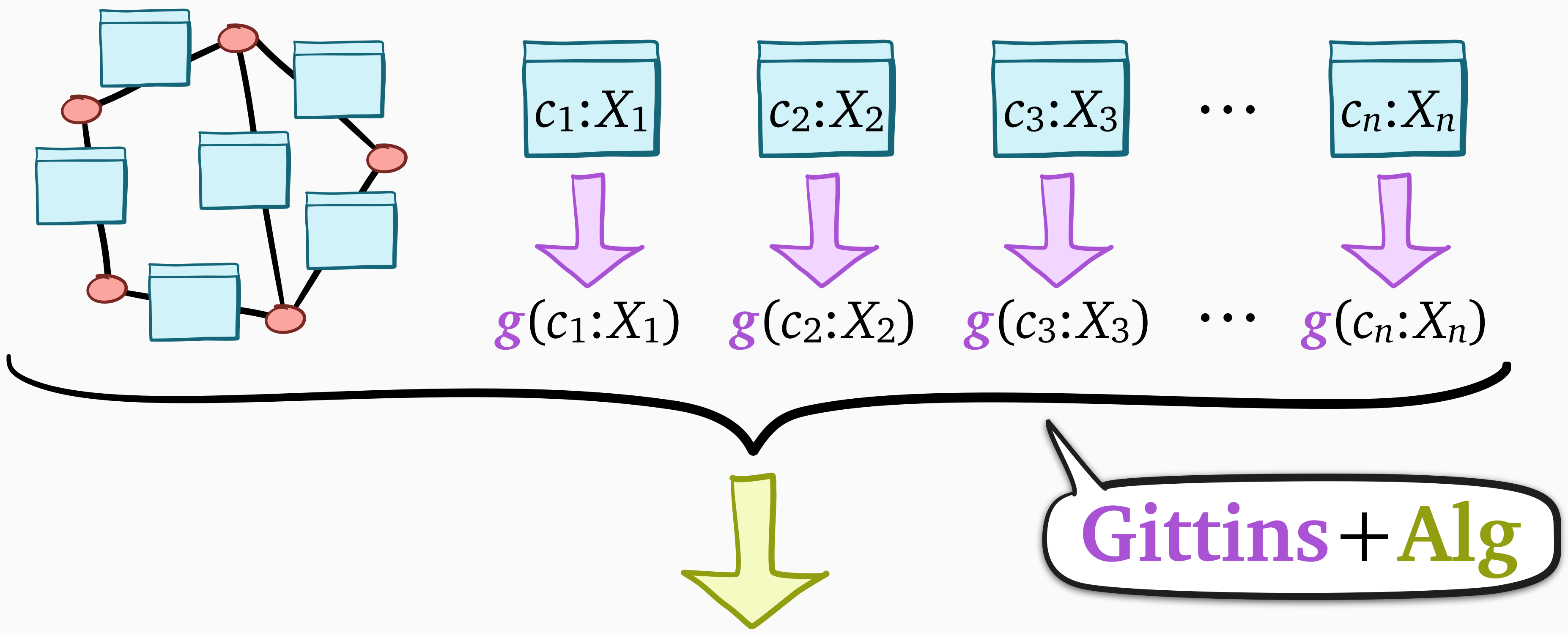
Deterministic algorithm:
"greedy" algorithm **Alg**

Classic problem



Deterministic algorithm:
pick smallest number

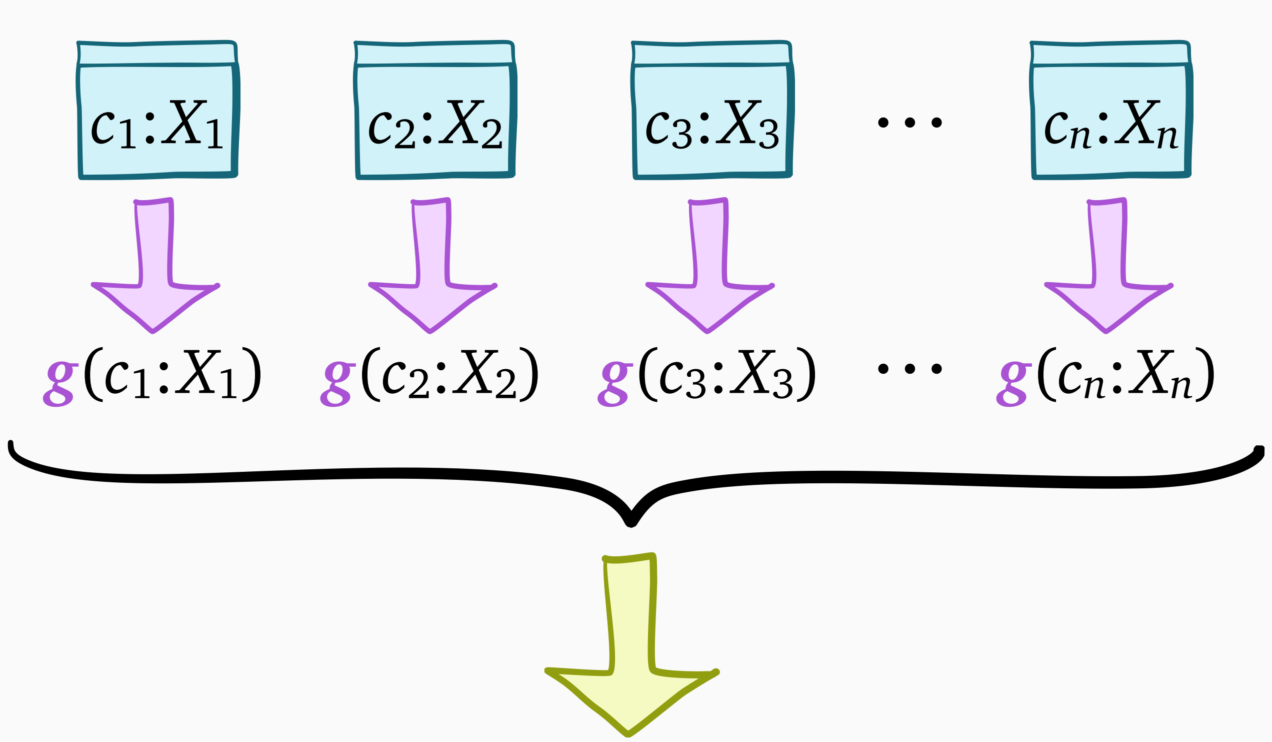
Combinatorial problems



Deterministic algorithm:
“greedy” algorithm **Alg**

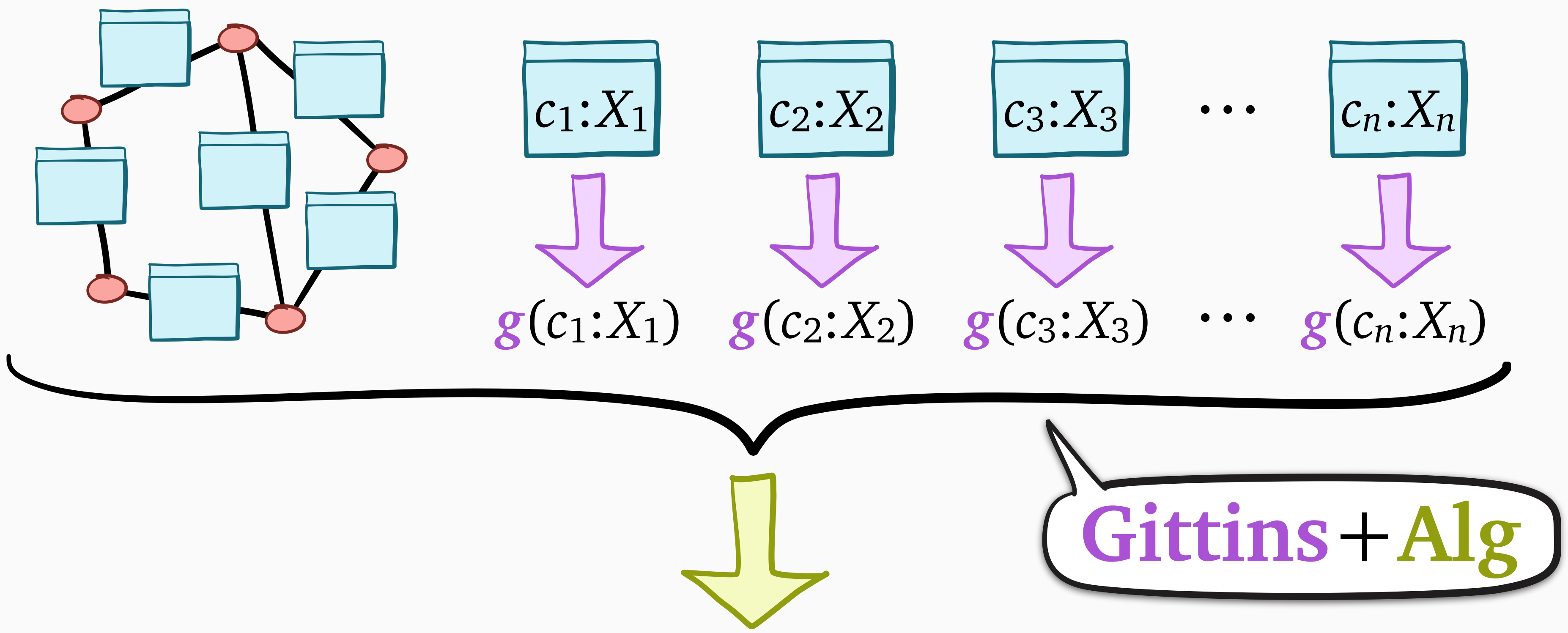
Theorem [Singla, 2018]**:** if **Alg** is a “greedy” algorithm, then **Gittins+Alg** and **Alg** have the same approximation ratio

Classic problem



Deterministic algorithm:
pick smallest number

Combinatorial problems

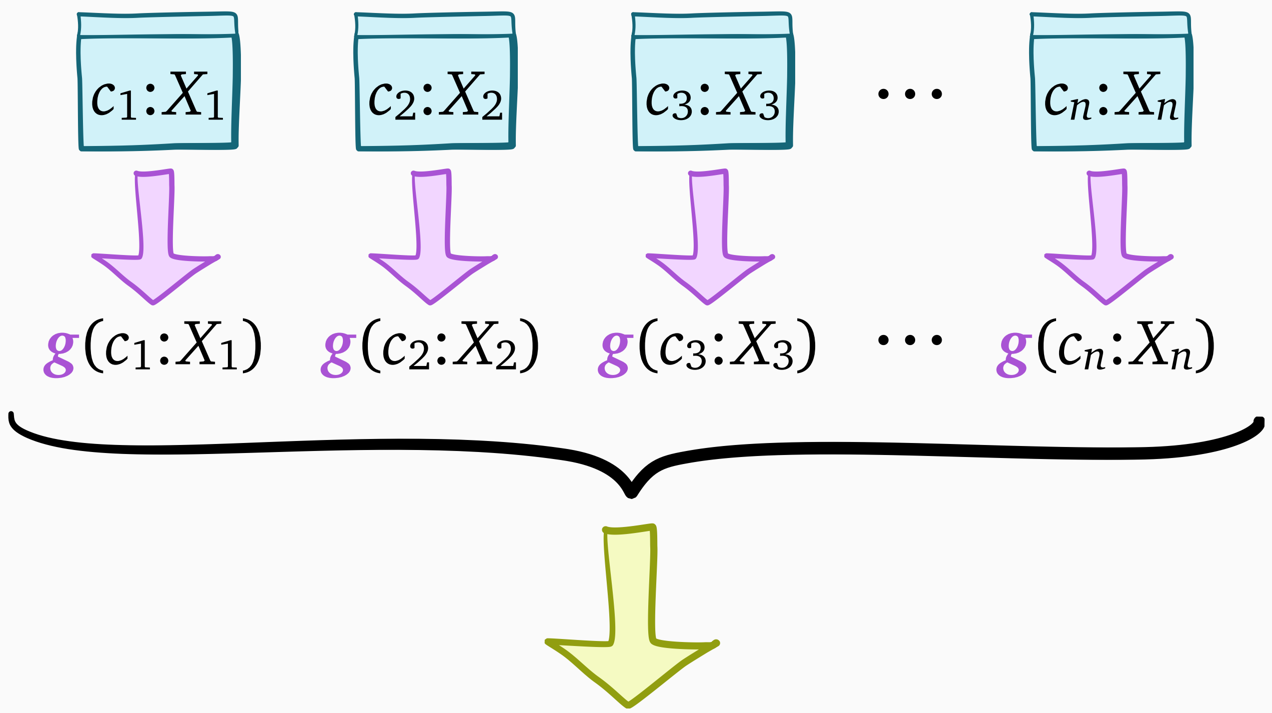


Deterministic algorithm:
"greedy" algorithm **Alg**

Theorem [Singla, 2018]: if **Alg** is a "greedy" algorithm, then **Gittins**+**Alg** and **Alg** have the same approximation ratio

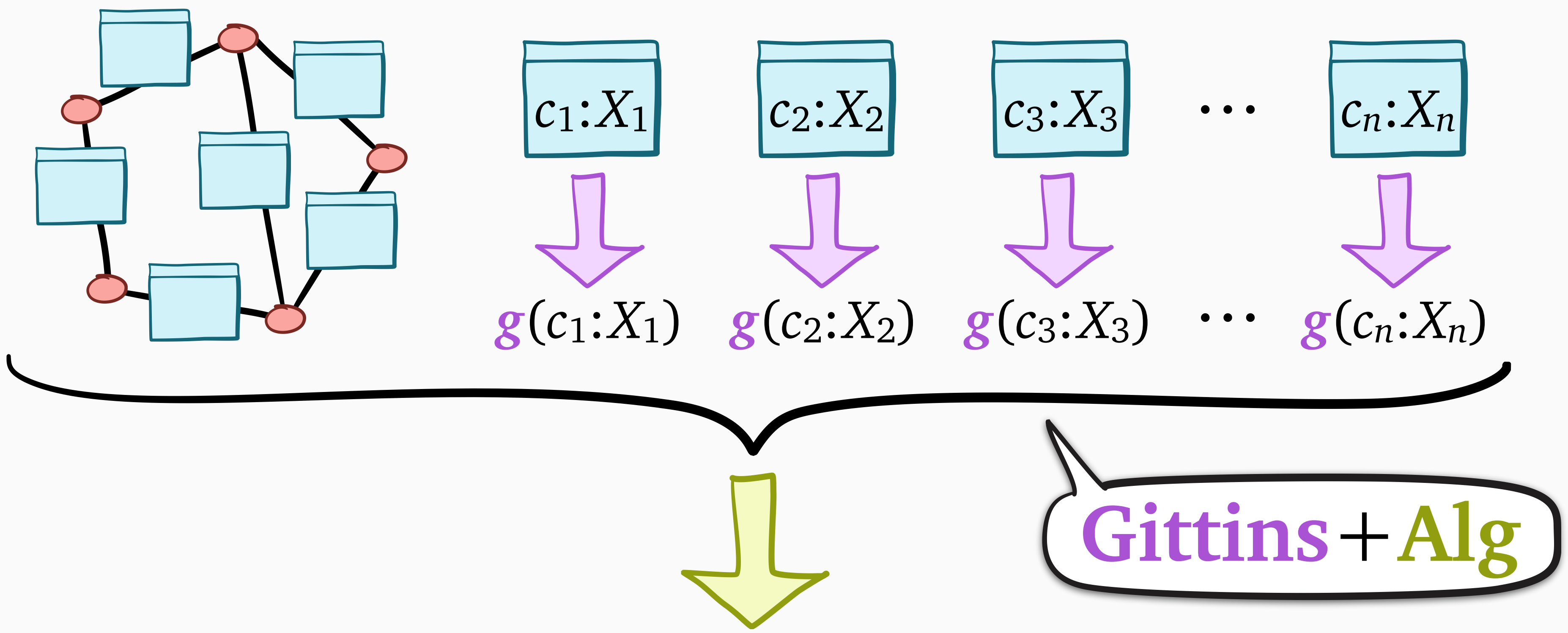
deterministic

Classic problem



Deterministic algorithm:
pick smallest number

Combinatorial problems



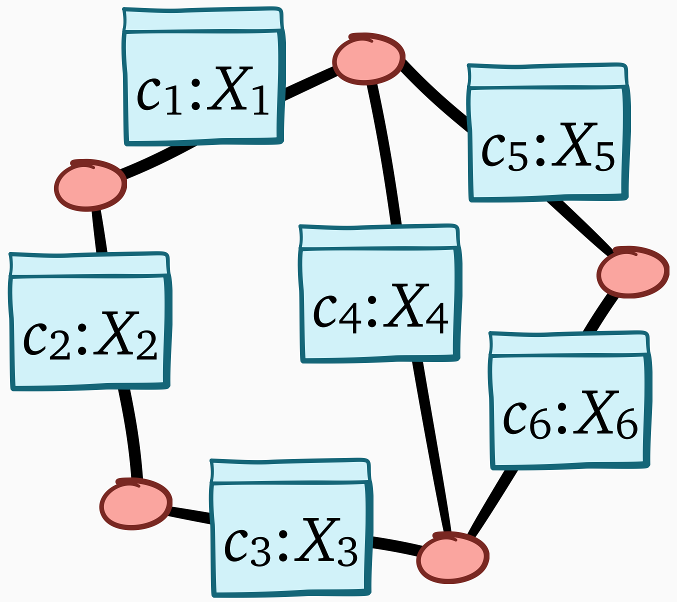
Deterministic algorithm:
“greedy” algorithm **Alg**

Theorem [Singla, 2018]: if **Alg** is a “greedy” algorithm, then **Gittins**+**Alg** and **Alg** have the same approximation ratio

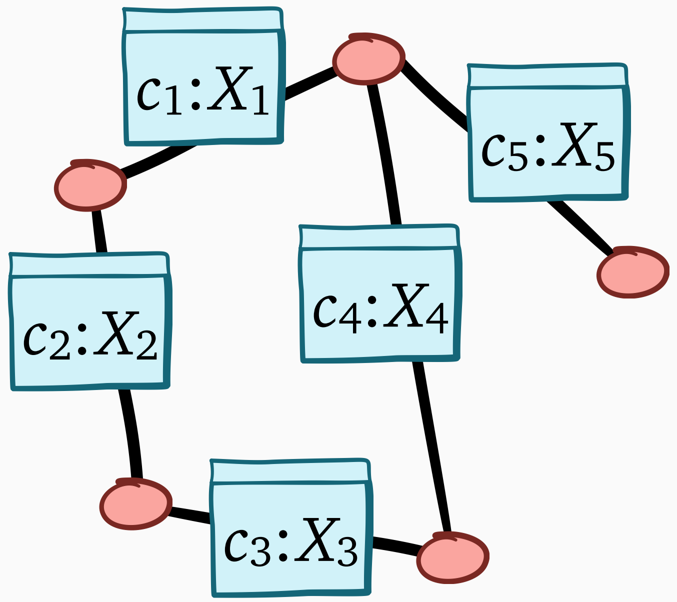
Pandora’s box

deterministic

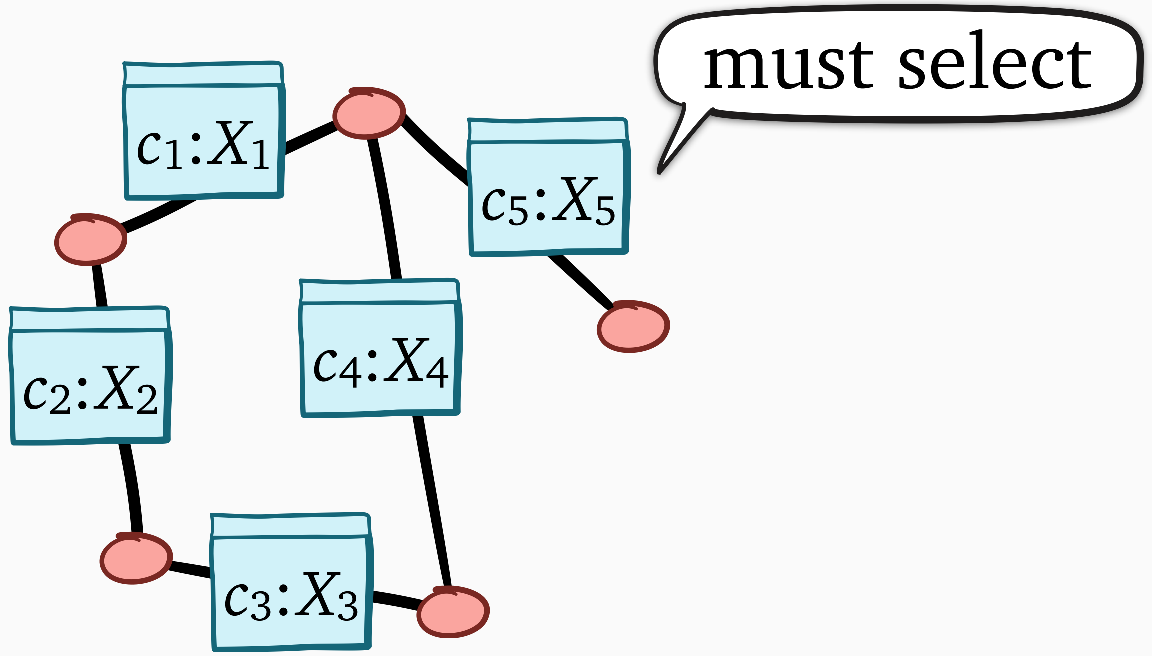
Example:
build a spanning tree



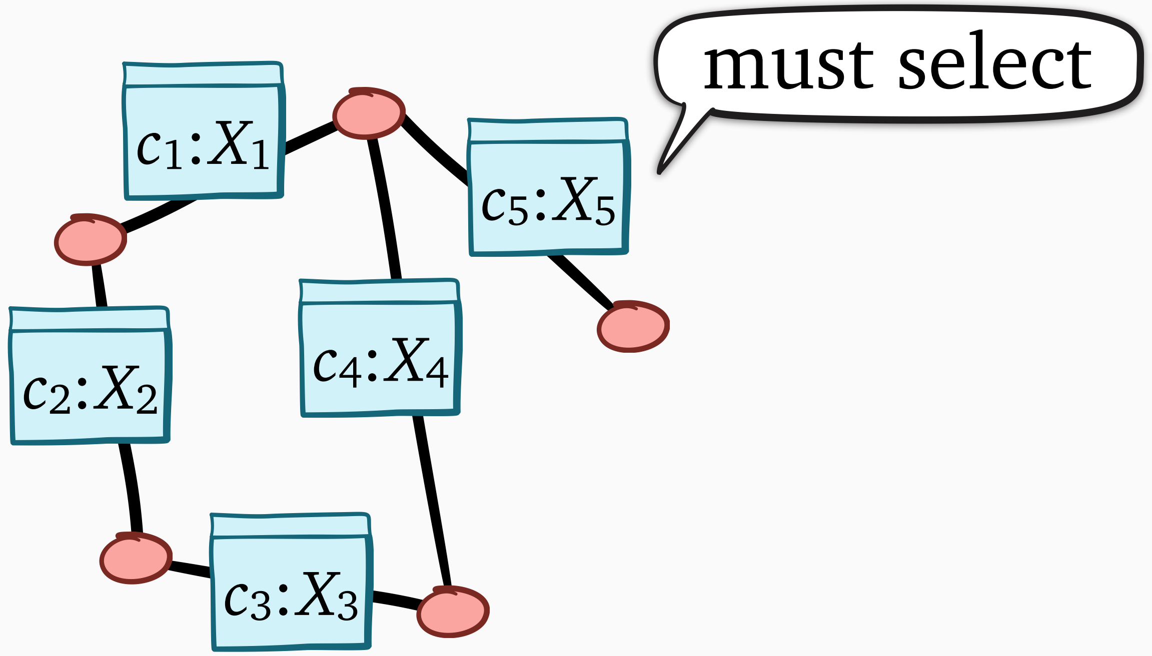
Example:
build a spanning tree



Example:
build a spanning tree



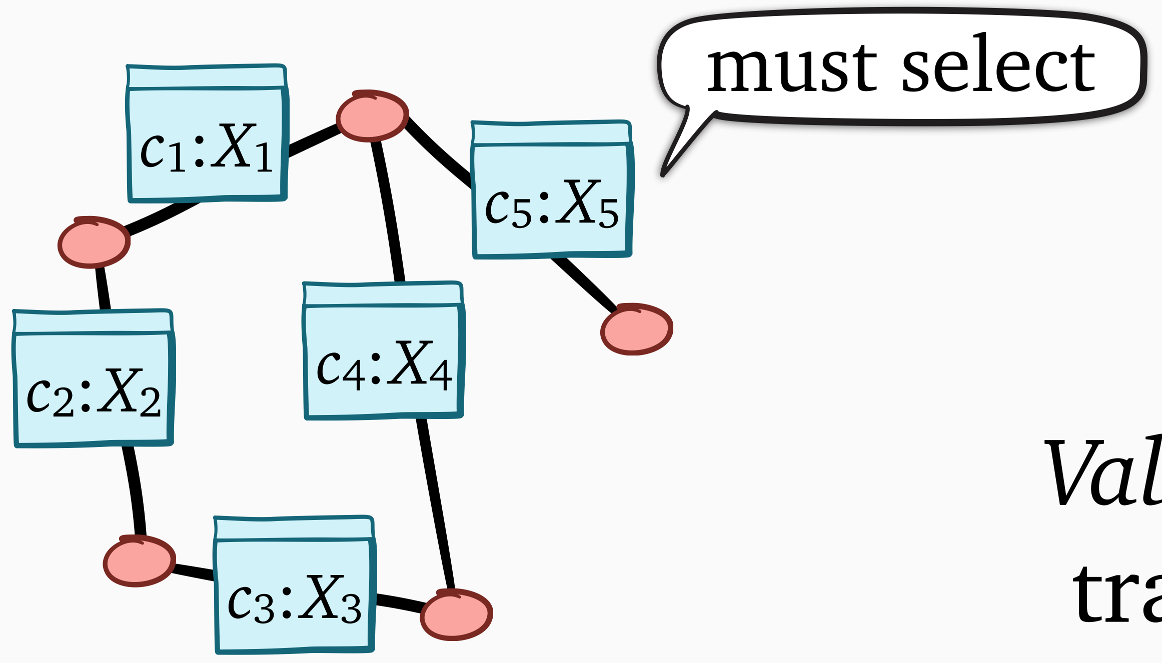
Example:
build a spanning tree



Optional inspection:

allow selecting closed boxes

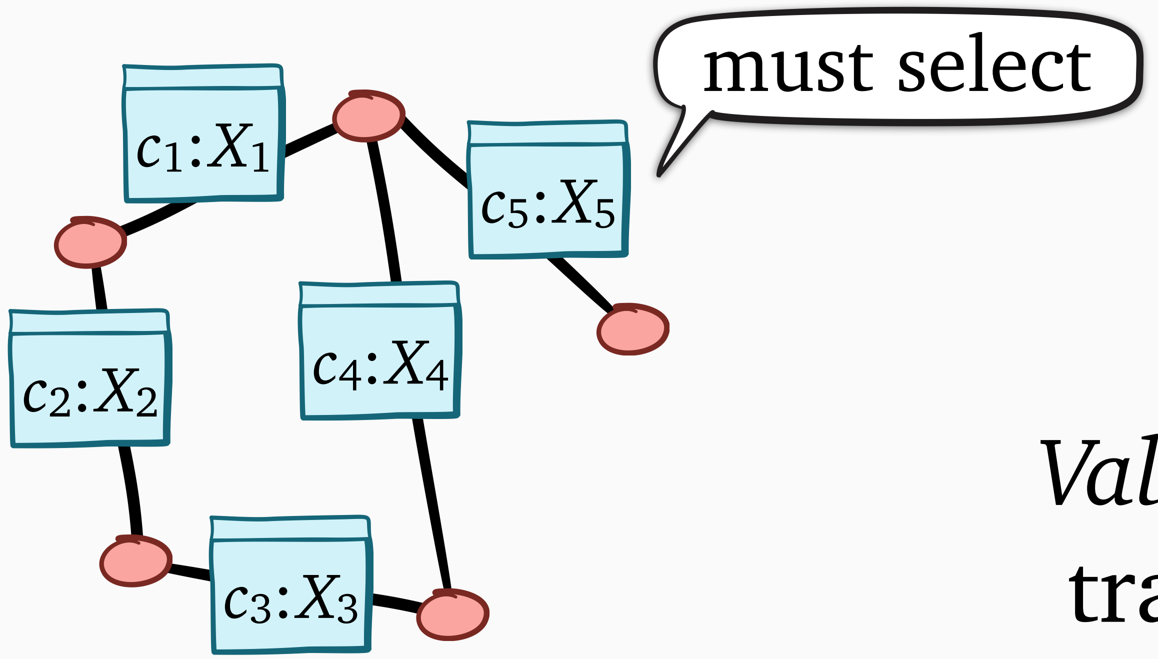
Example:
build a spanning tree



Optional inspection:
allow selecting closed boxes

Value of information:
tradeoff within box

Example:
build a spanning tree



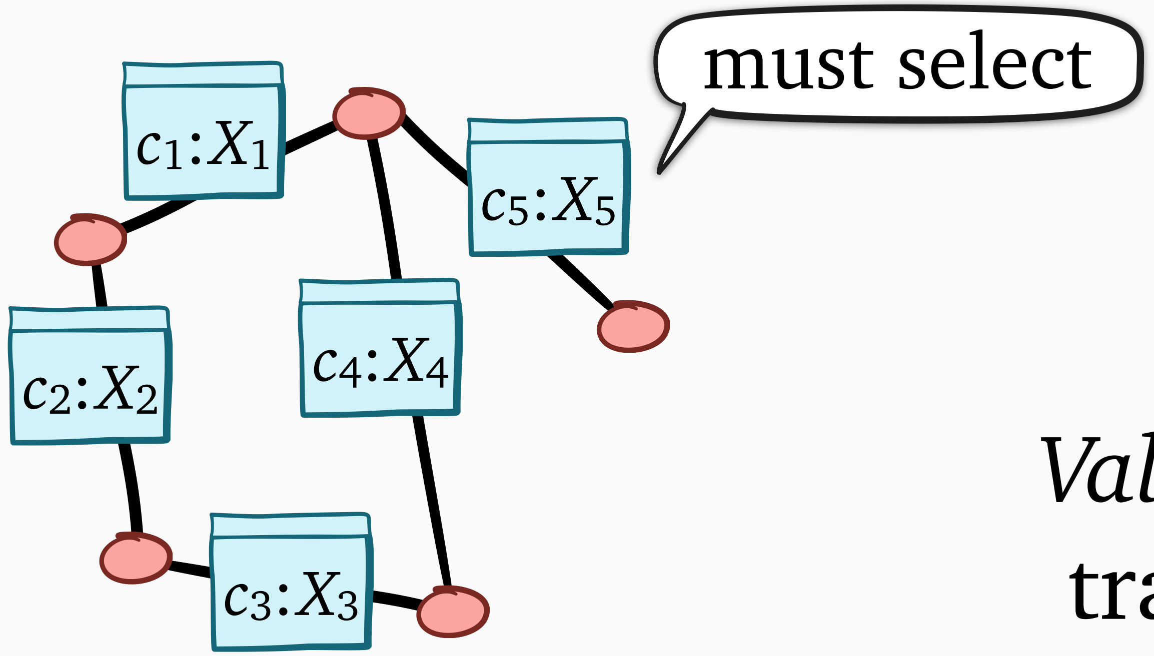
Optional inspection:

allow selecting closed boxes

Value of information:
tradeoff within box

Decompose hard problems:
more actions within box

Example:
build a spanning tree



Optional inspection:

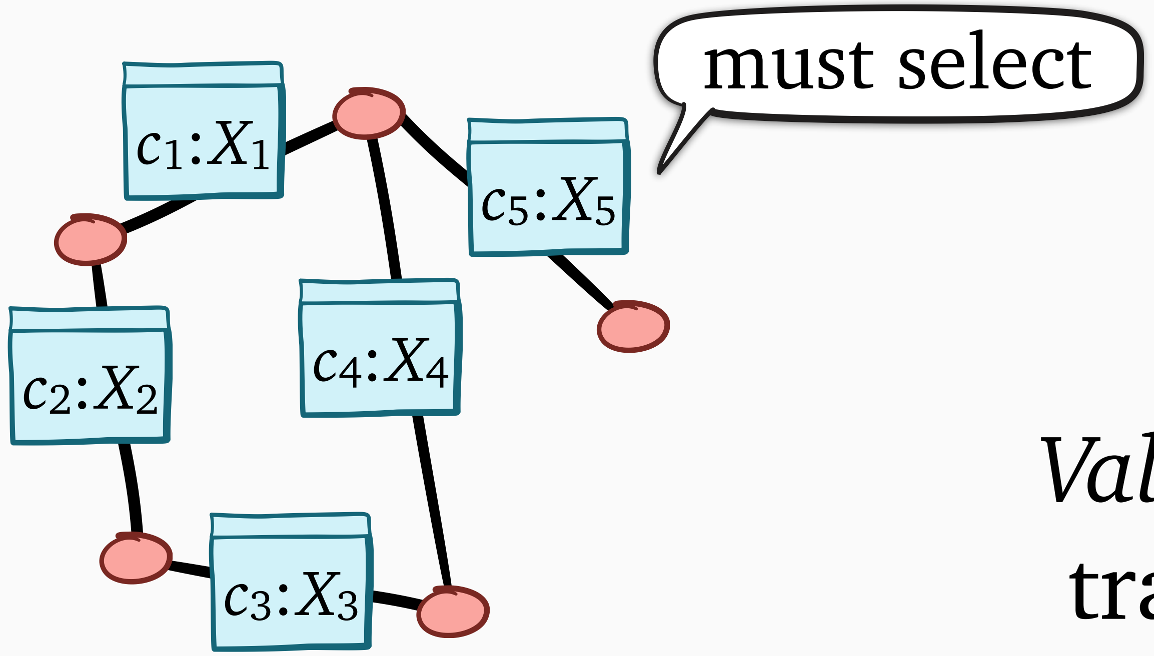
allow selecting closed boxes

Value of information:
tradeoff within box

Decompose hard problems:
more actions within box

Optional inspection is hard!

Example:
build a spanning tree



Optional inspection:

allow selecting closed boxes

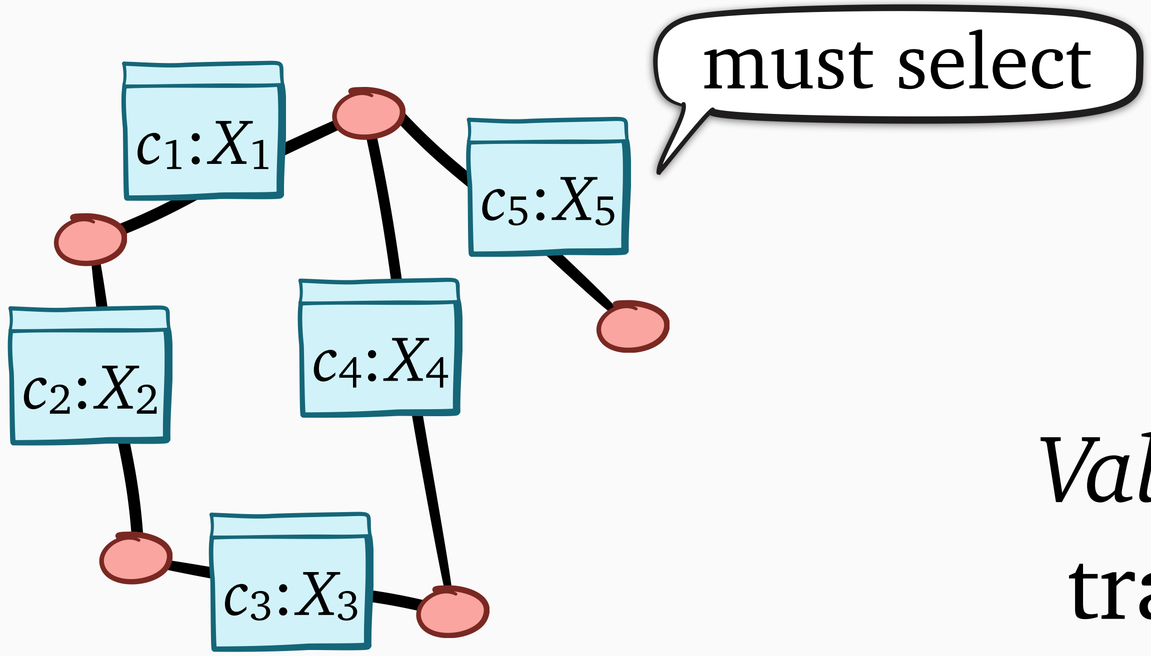
Value of information:
tradeoff within box

Decompose hard problems:
more actions within box

Optional inspection is hard!

- Gittins optimality doesn't generalize

Example:
build a spanning tree



Optional inspection:
allow selecting closed boxes

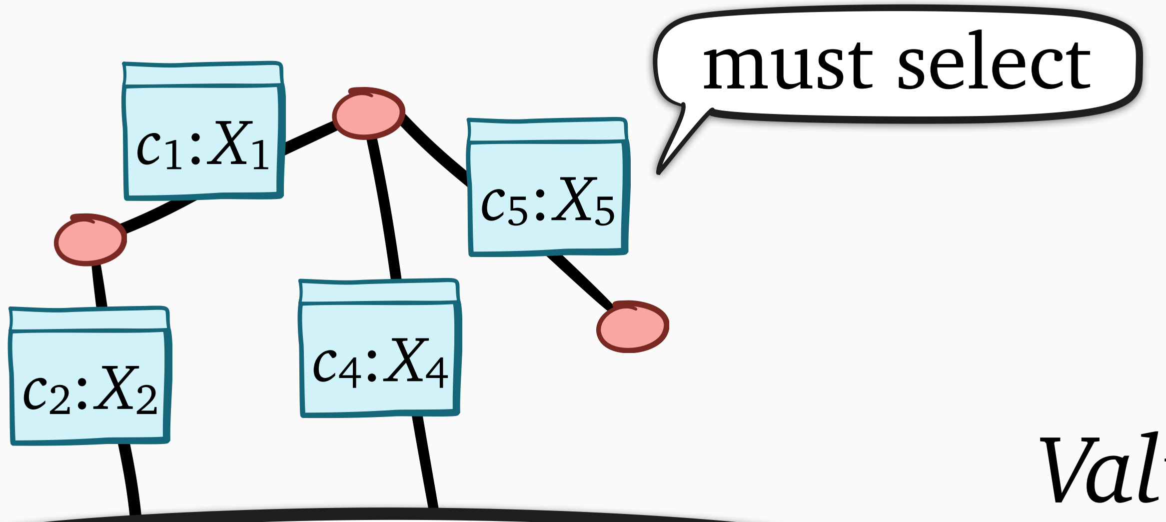
Value of information:
tradeoff within box

Decompose hard problems:
more actions within box

Optional inspection is hard!

- Gittins optimality doesn’t generalize
- Progress on classic problem, but combinatorial problems open

Example:
build a spanning tree



$1-1/e$ [Beyhaghi & Kleinberg, 2019]
4/5 [Guha, Munagala, & Sarkar, 2008]
PTAS [Fu, Li, & Liu, 2023]
PTAS [Beyhaghi & Cai, 2023]

Optional inspection:

allow selecting closed boxes

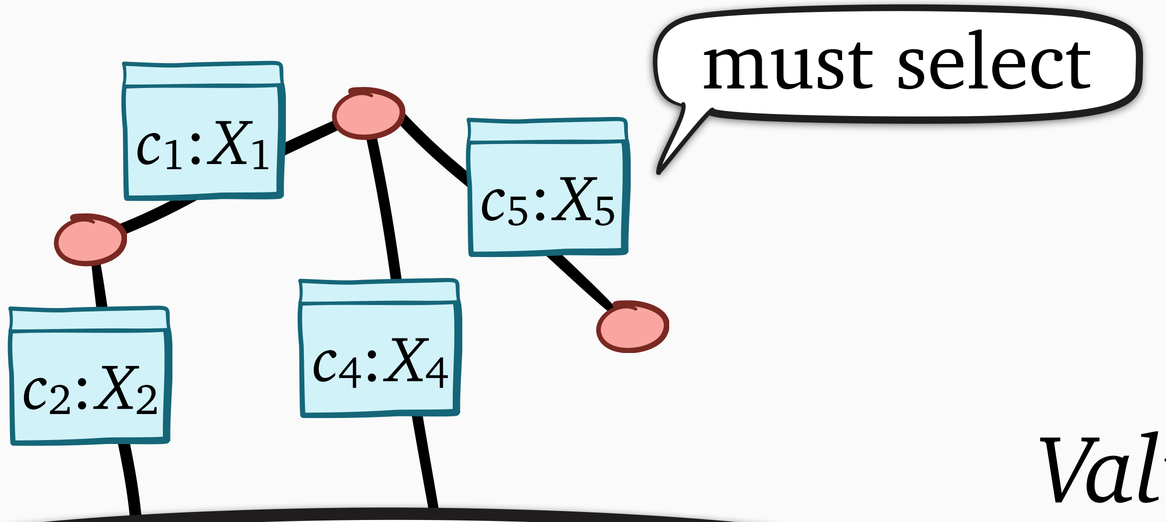
Value of information:
tradeoff within box

Decompose hard problems:
more actions within box

ptional inspection is hard!

- Gittins optimality doesn't generalize
- Progress on classic problem, but combinatorial problems open

Example:
build a spanning tree



$1-1/e$ [Beyhaghi & Kleinberg, 2019]
$4/5$ [Guha, Munagala, & Sarkar, 2008]
PTAS [Fu, Li, & Liu, 2023]
PTAS [Beyhaghi & Cai, 2023]

Optional inspection:

allow selecting closed boxes

Value of information:
tradeoff within box

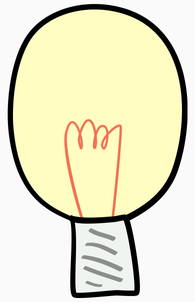
Decompose hard problems:
more actions within box

Gittins-y policy works in special cases [Doval, 2018]

Optional inspection is hard:

- Gittins optimality doesn't generalize
- Progress on classic problem, but combinatorial problems open

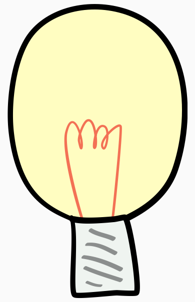
Our contribution



Local Hedging (LH)

New *decomposition-based* technique for optional inspection

Our contribution

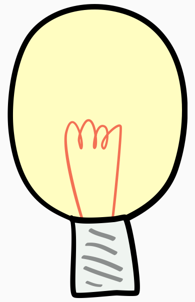


Local Hedging (**LH**)

New *decomposition-based* technique for optional inspection

- Reduces problem to required-inspection case

Our contribution

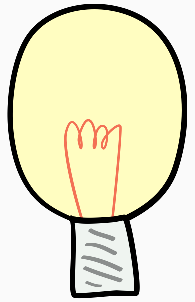


Local Hedging (LH)

New *decomposition-based* technique for optional inspection

- Reduces problem to required-inspection case
- Naturally generalizes to combinatorial problems

Our contribution



Local Hedging (LH)

New *decomposition-based* technique for optional inspection

- Reduces problem to required-inspection case
- Naturally generalizes to combinatorial problems

Theorem: if Alg is a “greedy” algorithm, then the approximation ratio of Gittins+Alg+LH is ≤ 4/3 times that of Alg

Our contribution

Local Hedging (LH)

New *decomposition-based* technique for optional inspection

- Reduces problem to required-inspection case
- Naturally generalizes to combinatorial problems

NEW!

Theorem: if Alg is a "greedy" algorithm, then the approximation ratio of Gittins+Alg+LH is ≤ 4/3 times that of Alg

price of reduction

Rest of this talk

Why does **Gittins** work under required inspection?

Rest of this talk

Why does **Gittins** work under required inspection?

What goes wrong under optional inspection?

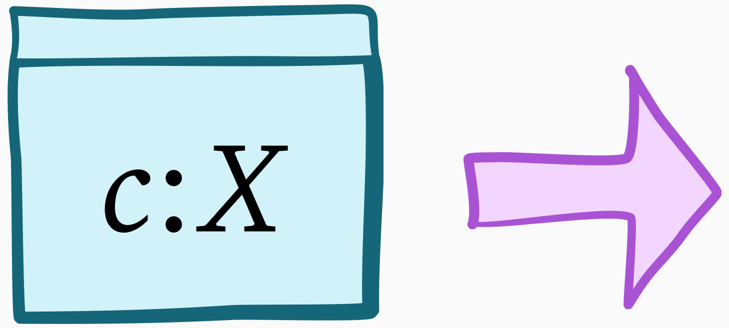
Rest of this talk

Why does **Gittins** work under required inspection?

What goes wrong under optional inspection?

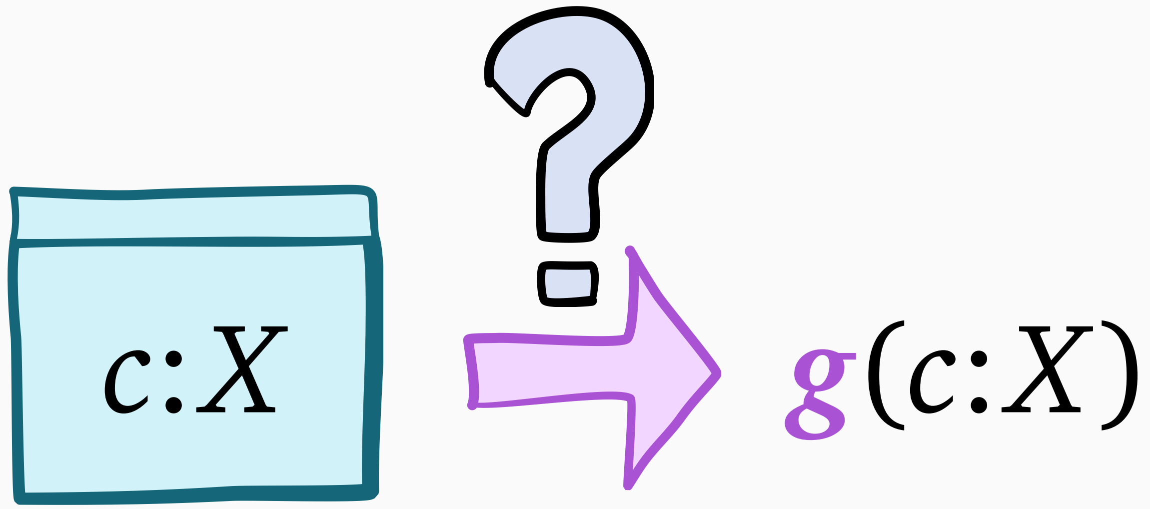
How does **Local Hedging** fix the problem?

Defining the **Gittins** index

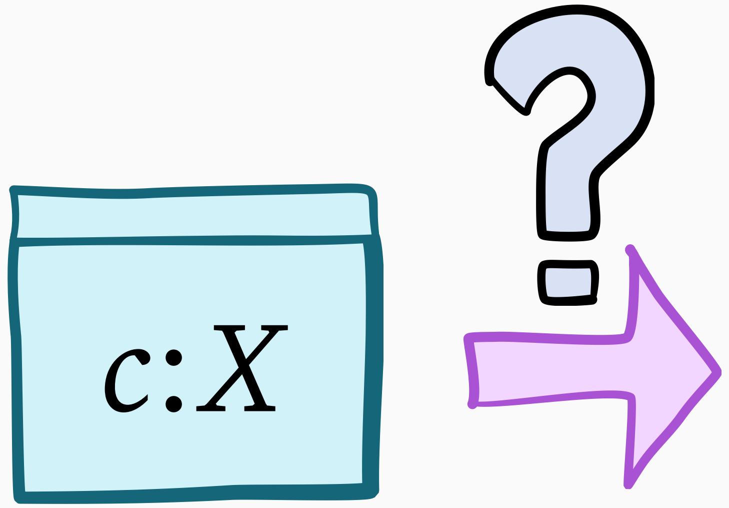


$g(c{:}X)$

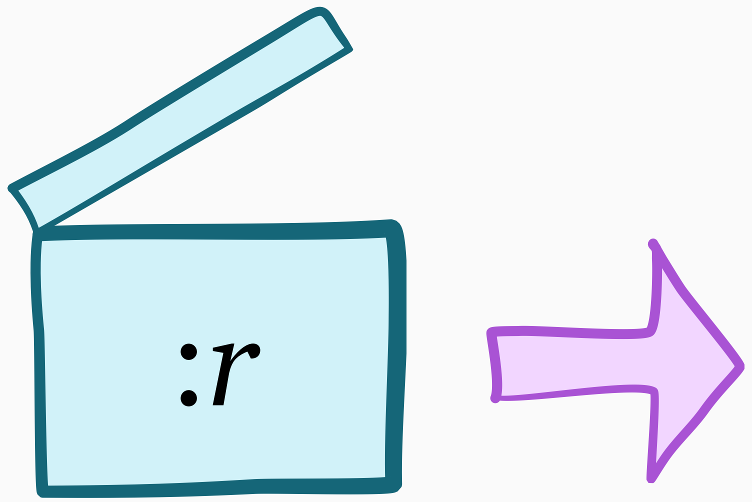
Defining the **Gittins** index



Defining the **Gittins** index

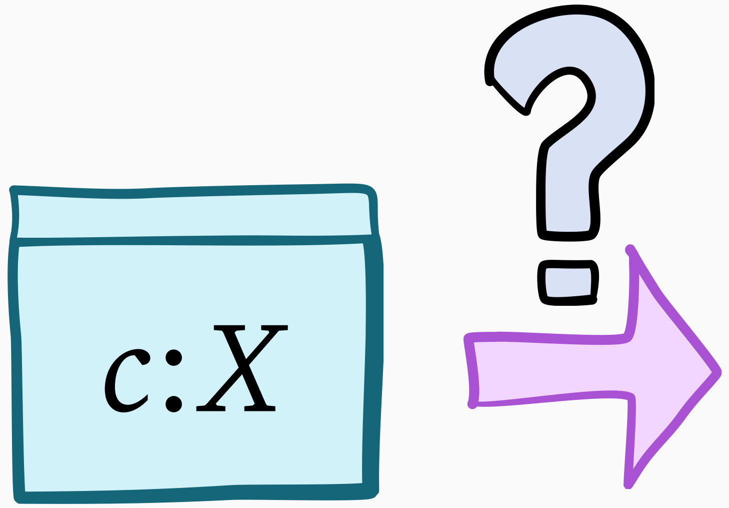


$\mathbf{g}(c{:}X)$

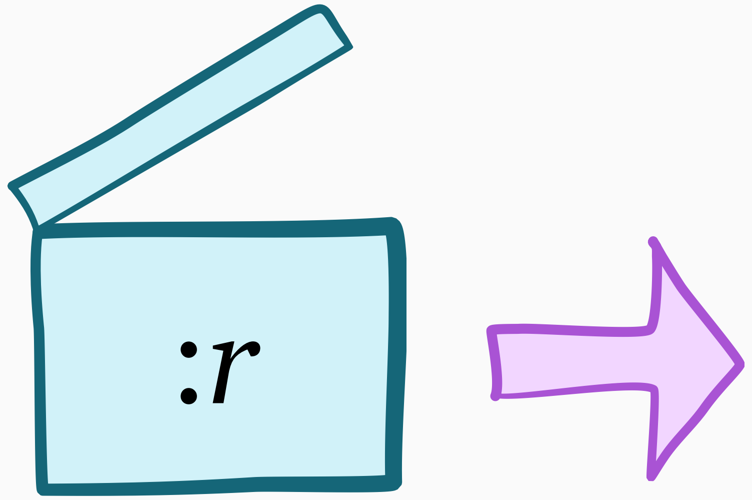


$\mathbf{g}({:}r) = r$

Defining the Gittins index

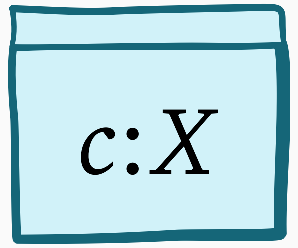


$g(c:X)$

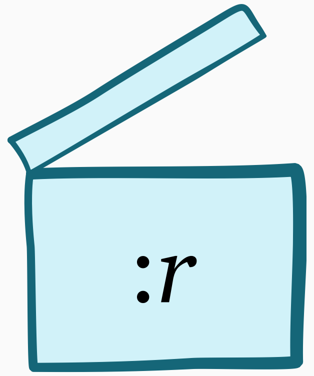


$g(:r) = r$

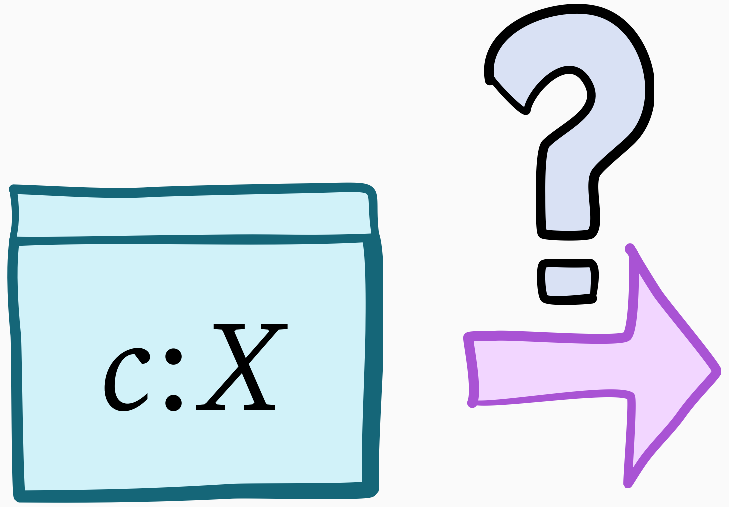
1.5-box problem



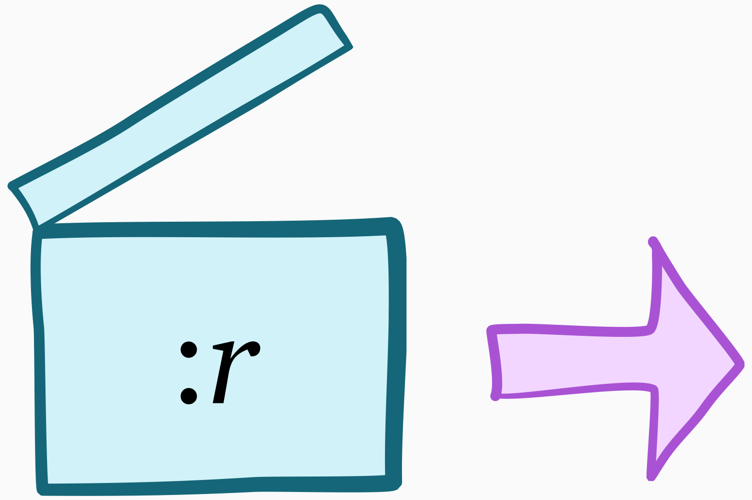
vs.



Defining the Gittins index

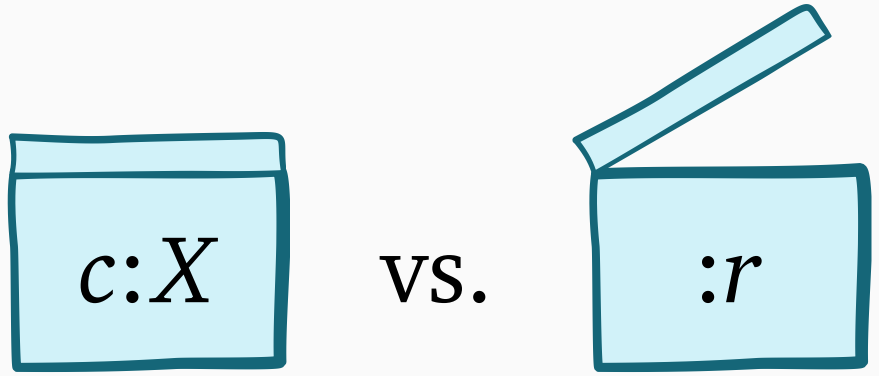


$g(c{:}X)$



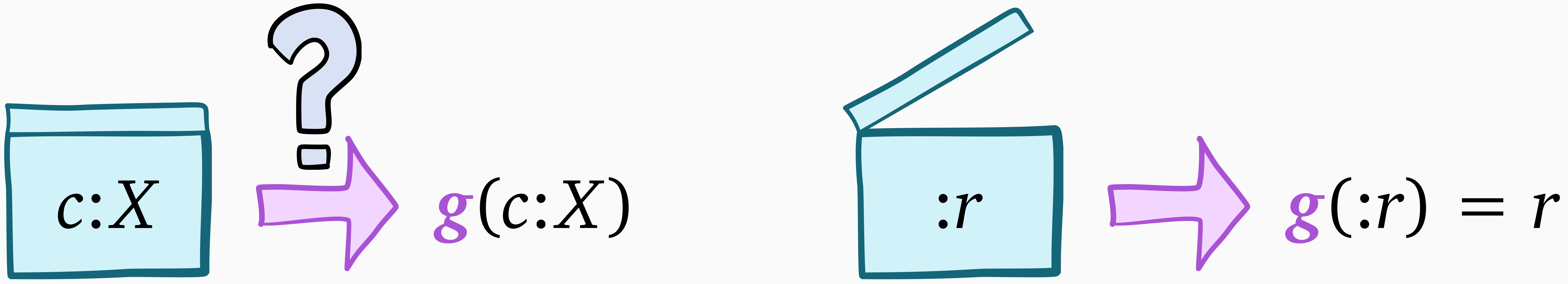
$g(:r) = r$

1.5-box problem

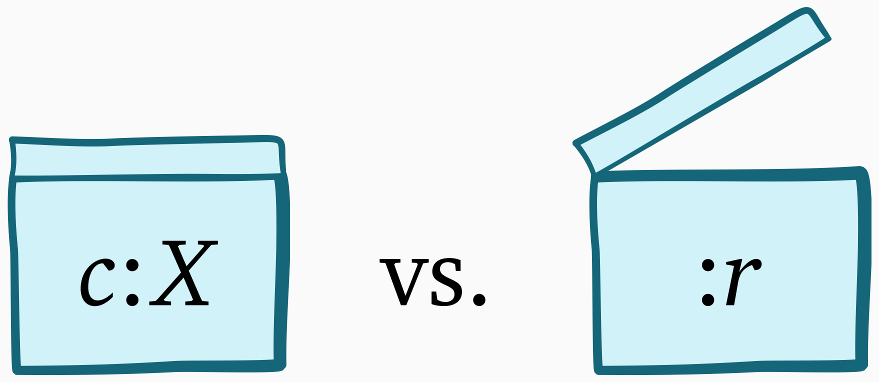


Key question: what to do in 1.5-box problem?

Defining the Gittins index

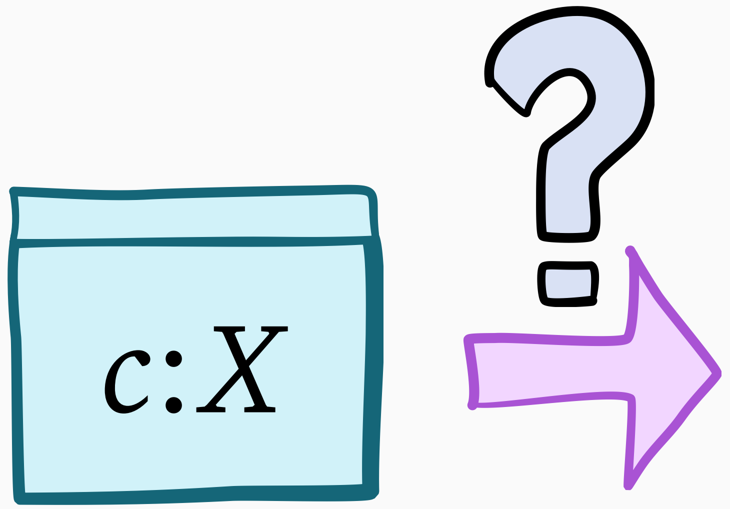


1.5-box problem

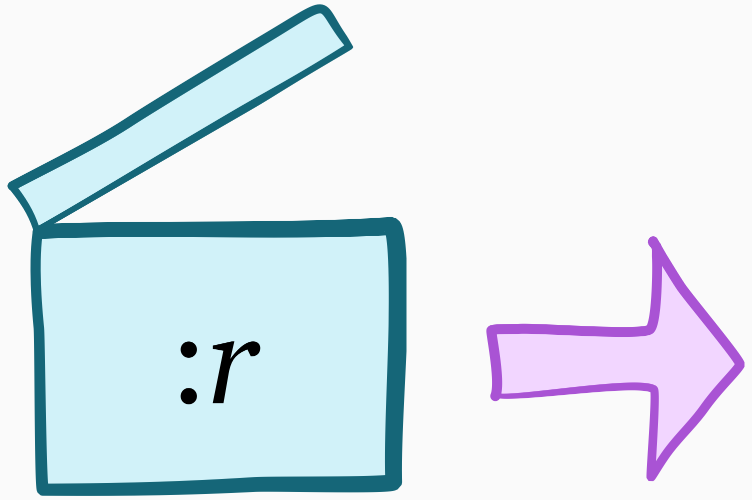


Key question: what to do in 1.5-box problem?

Defining the **Gittins** index

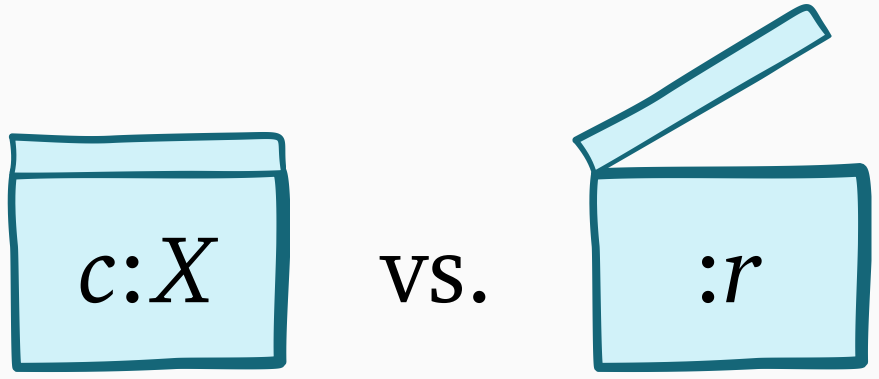


$g(c{:}X)$

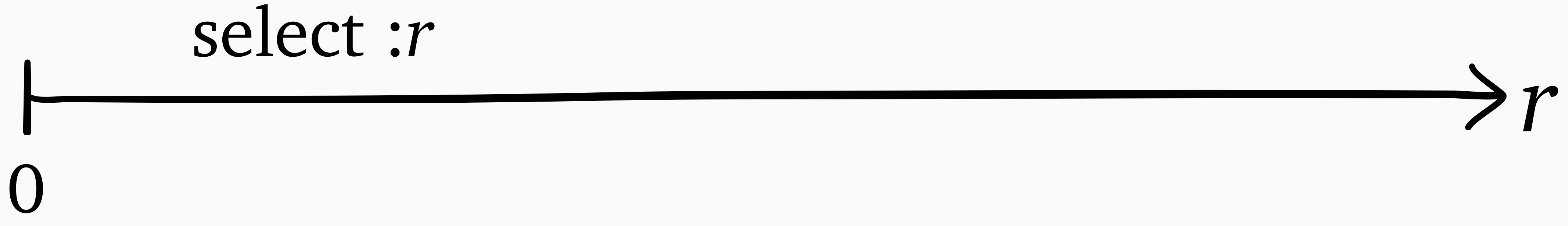


$g({:}r) = r$

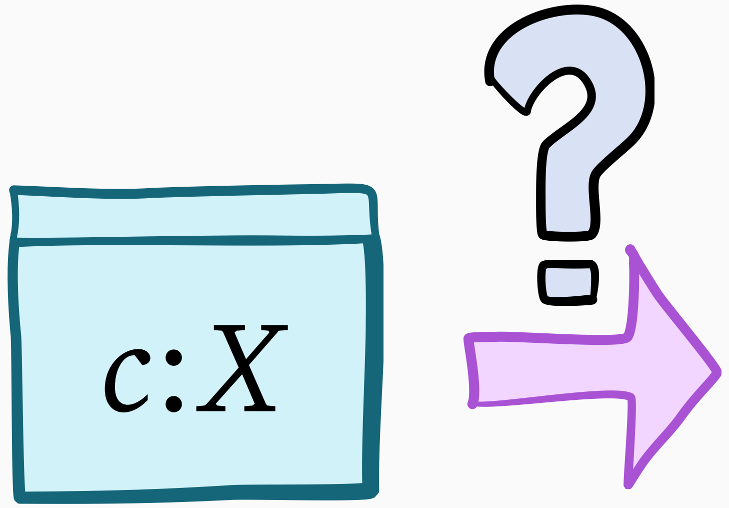
1.5-box problem



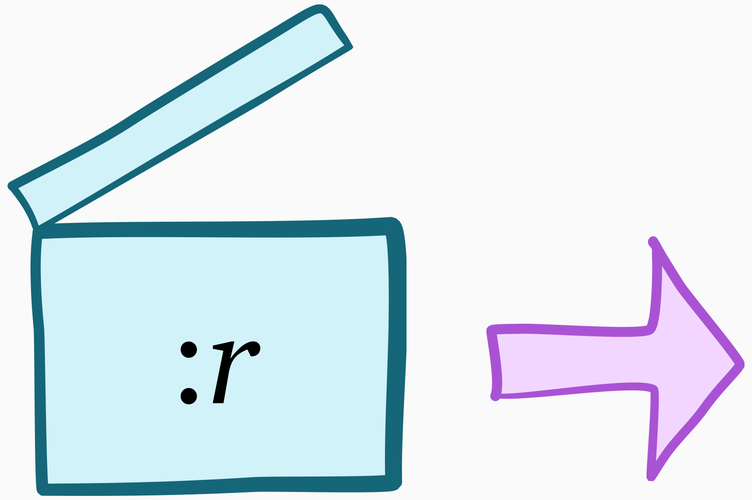
Key question: what to do in 1.5-box problem?



Defining the **Gittins** index

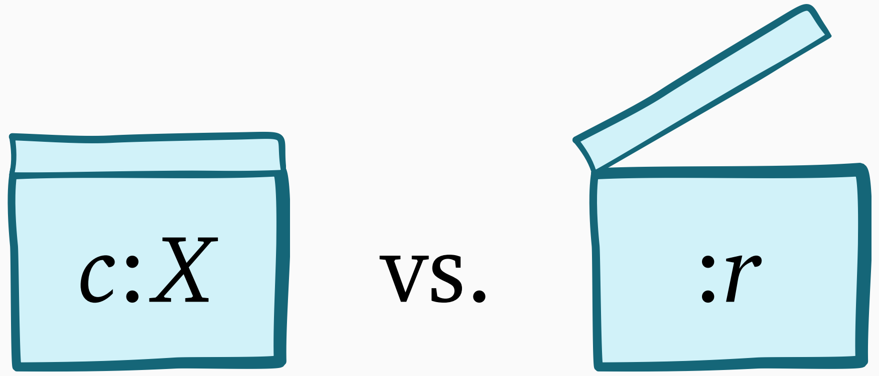


$g(c{:}X)$

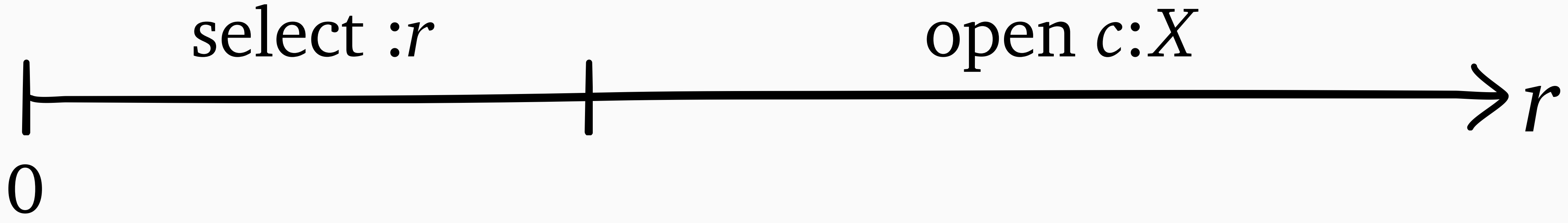


$g({:}r) = r$

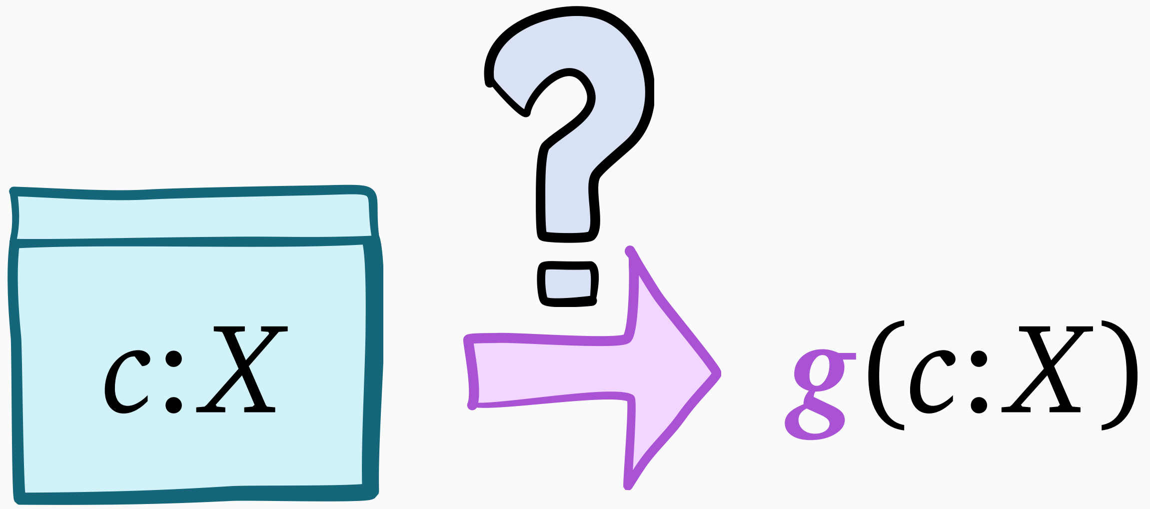
1.5-box problem



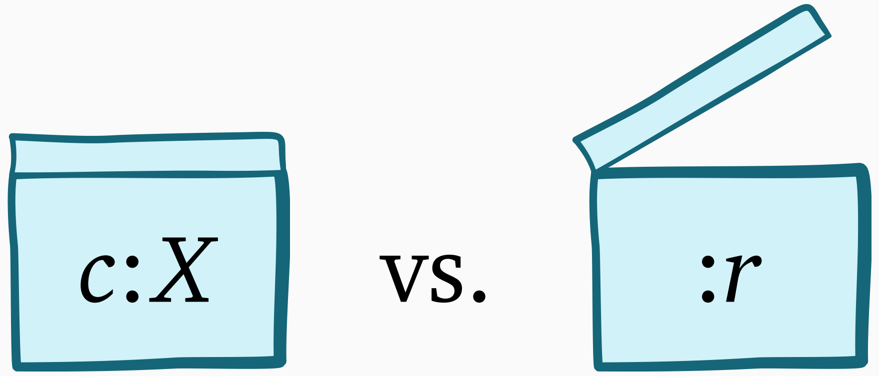
Key question: what to do in 1.5-box problem?



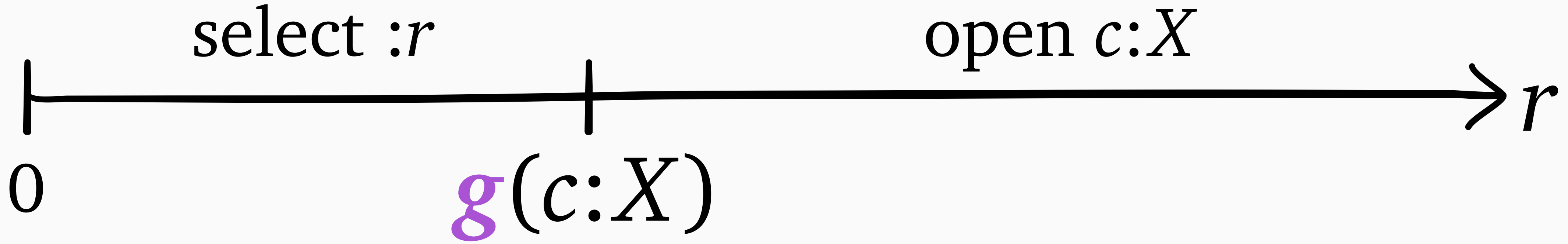
Defining the Gittins index



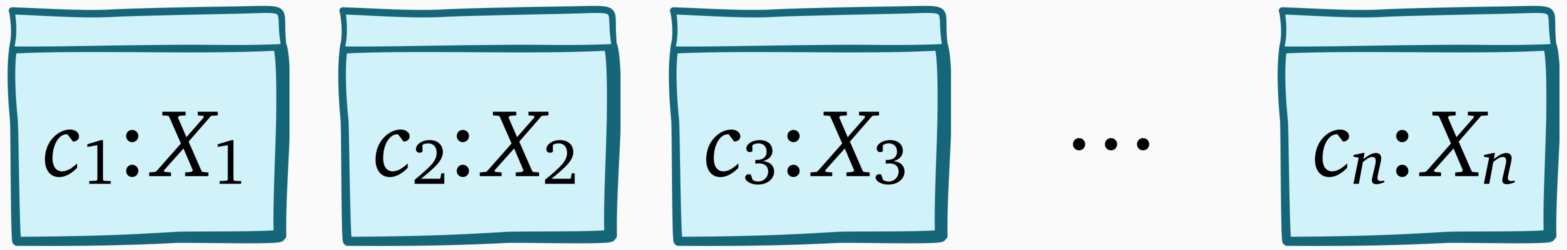
1.5-box problem



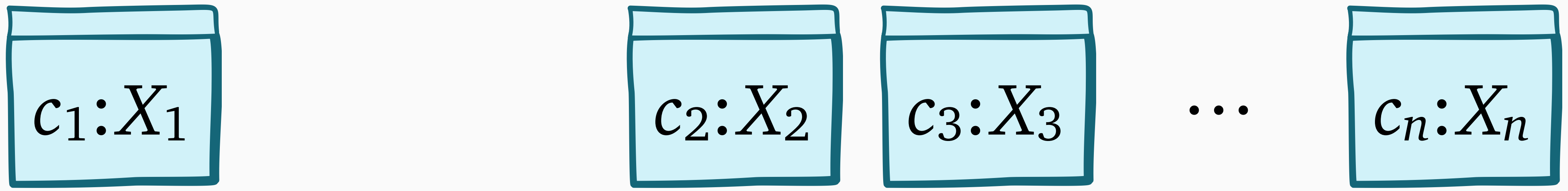
Key question: what to do in 1.5-box problem?



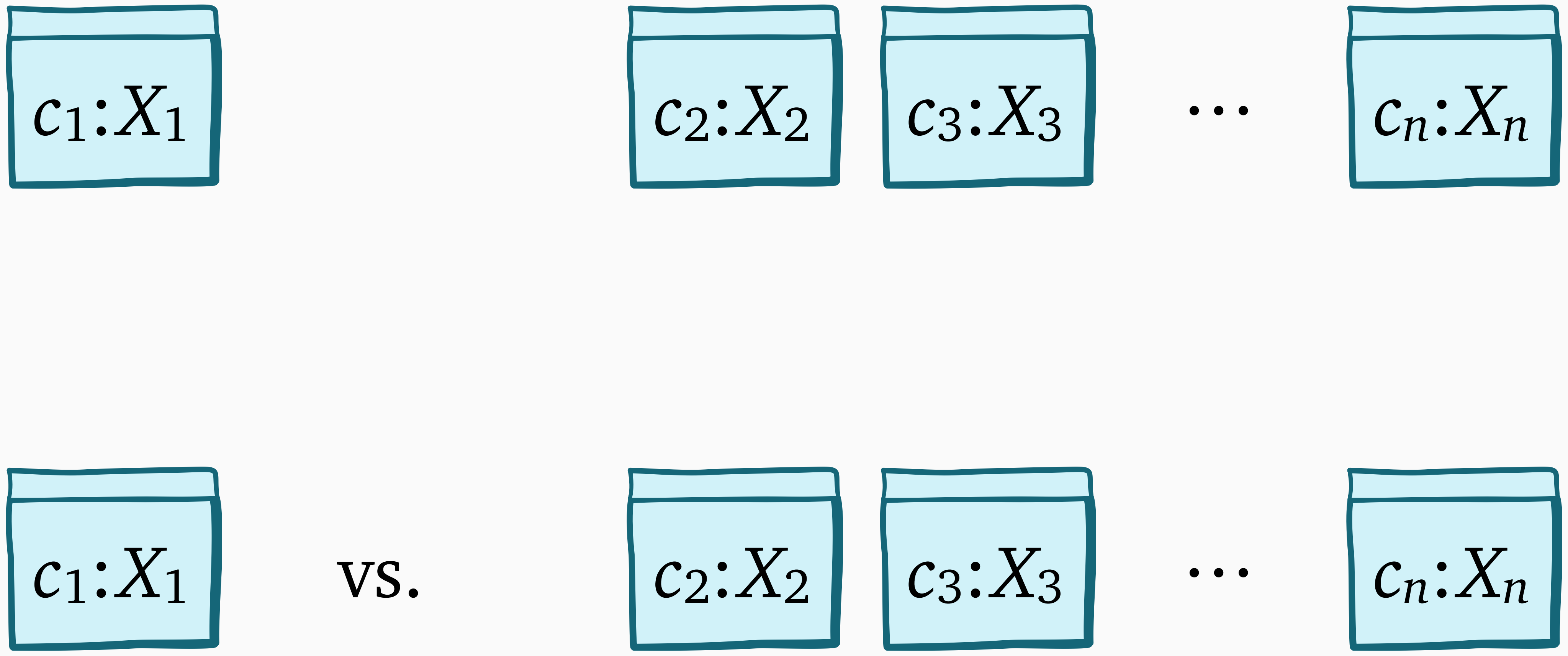
Why Gittins works under required inspection



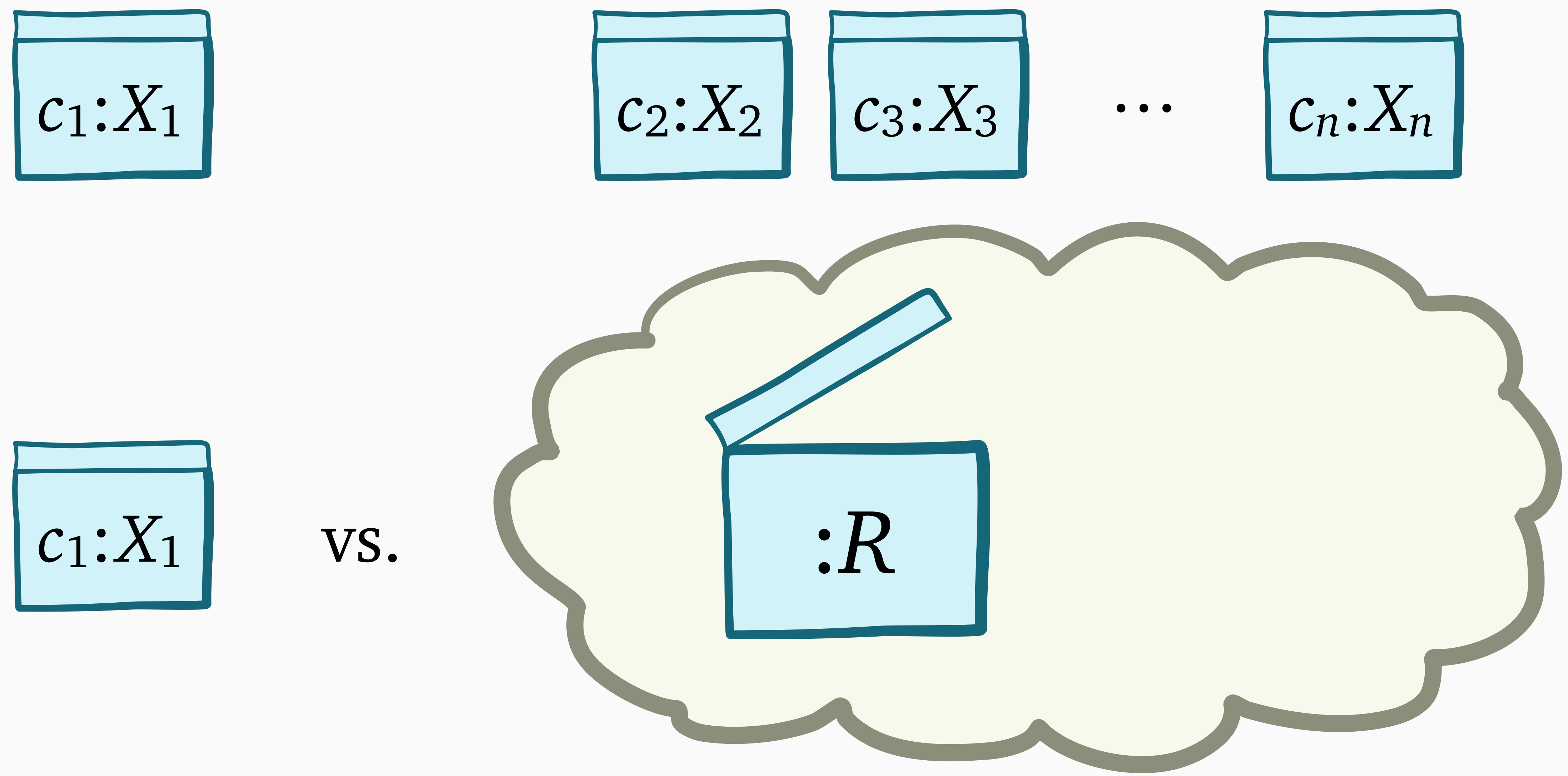
Why **Gittins** works under required inspection



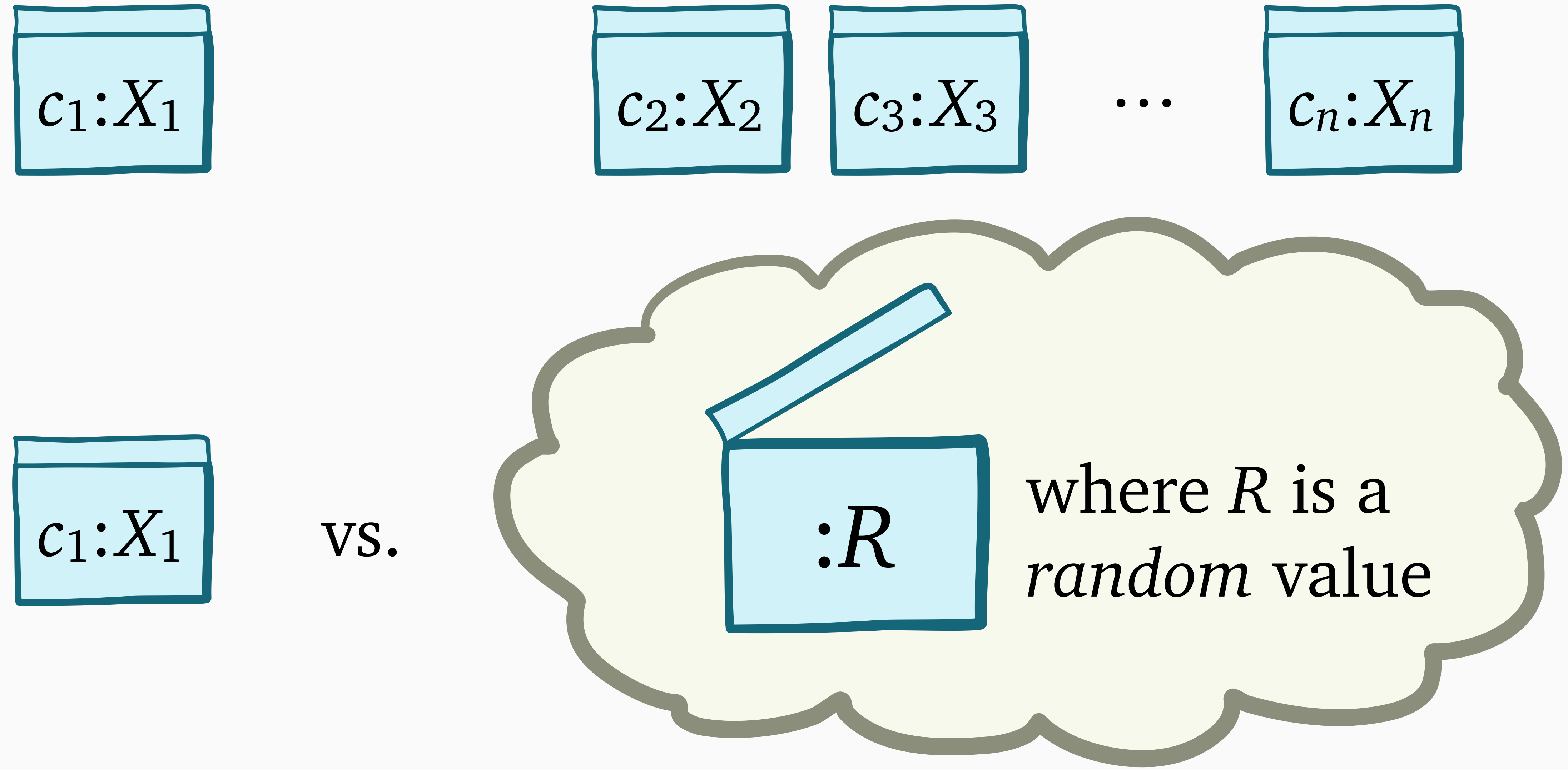
Why Gittins works under required inspection



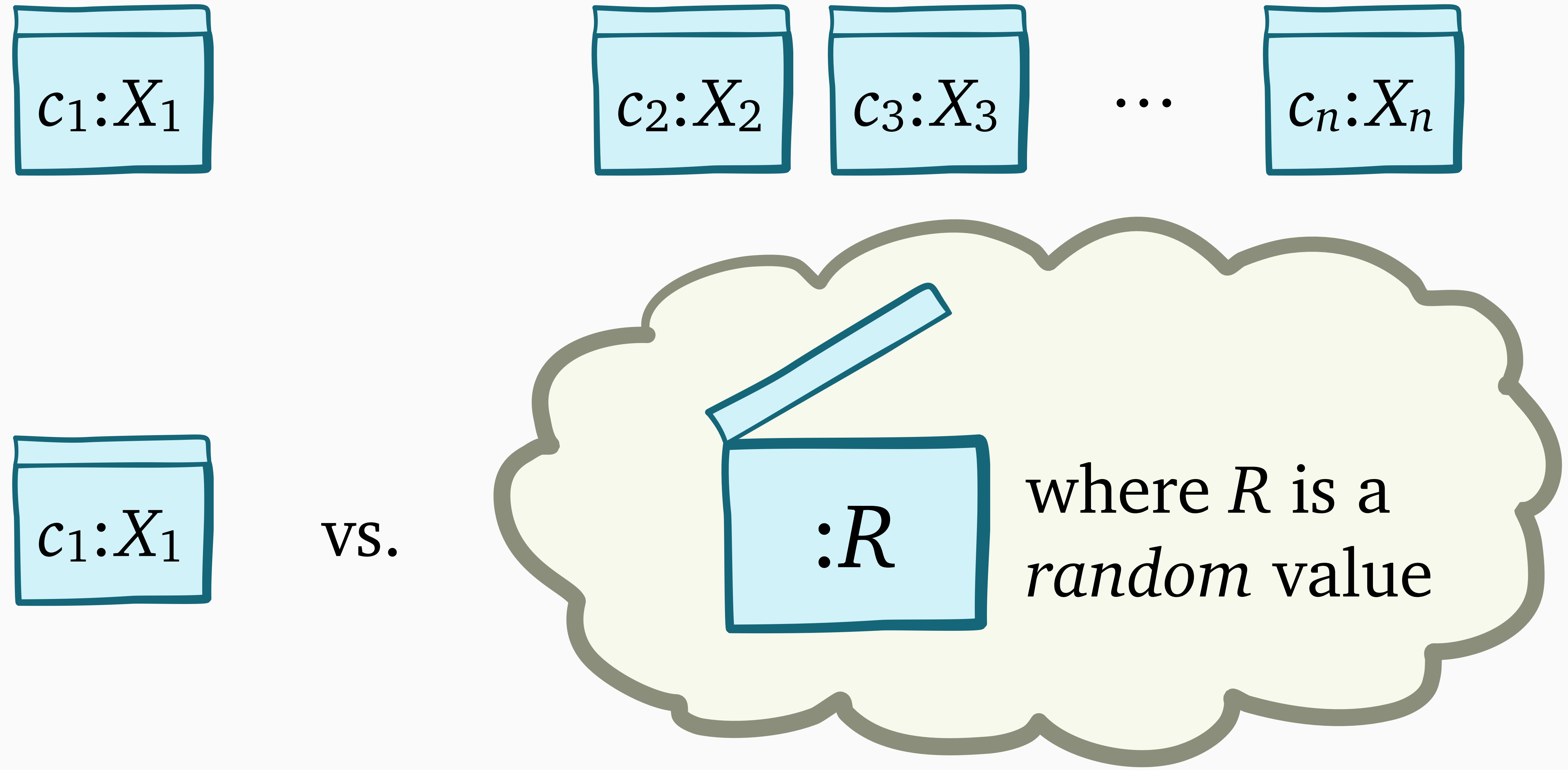
Why **Gittins** works under required inspection



Why Gittins works under required inspection

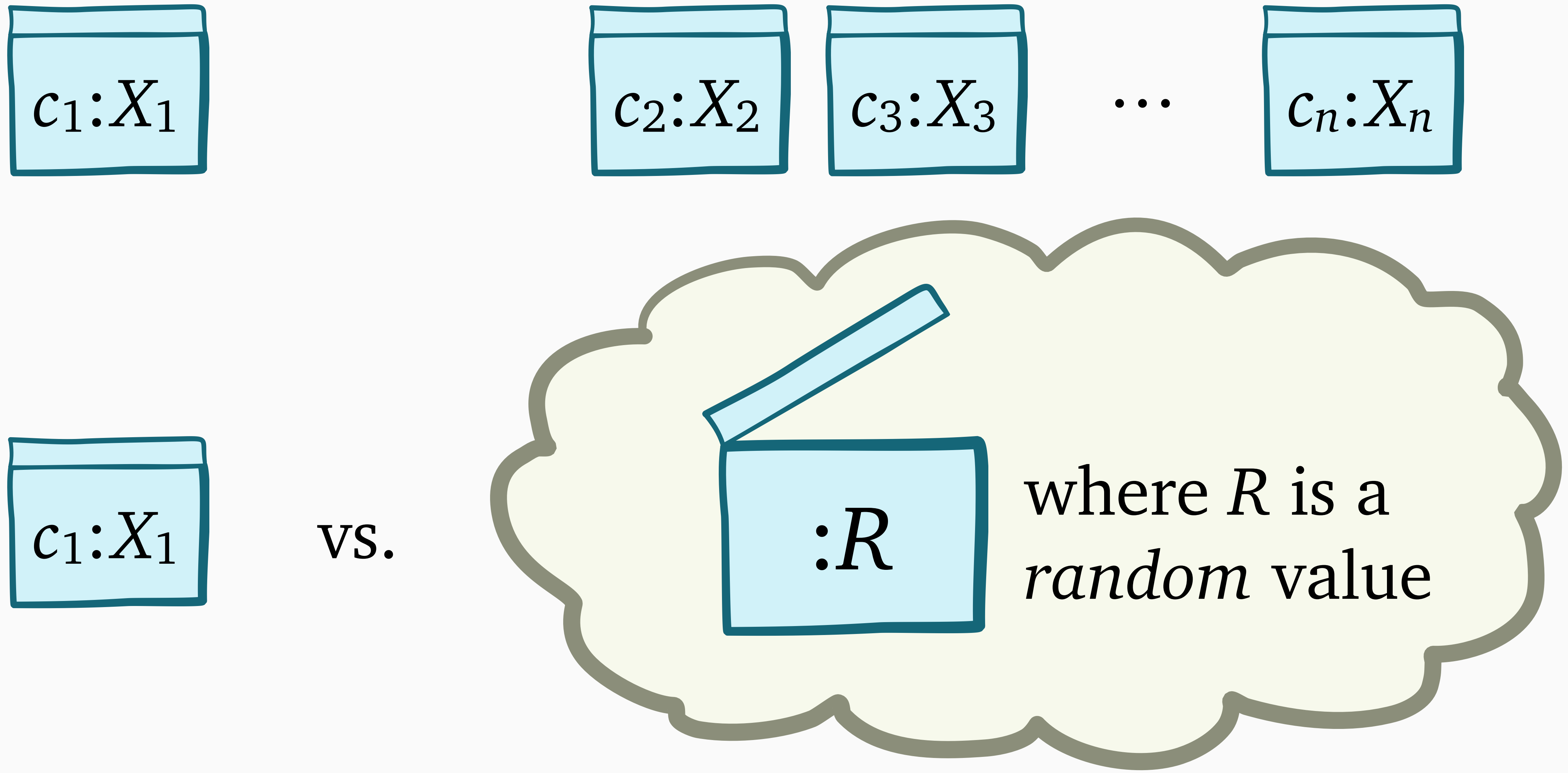


Why Gittins works under required inspection



Key property: $R \geq \min\{g(c_2{:}X_2), \ldots, g(c_n{:}X_n)\}$

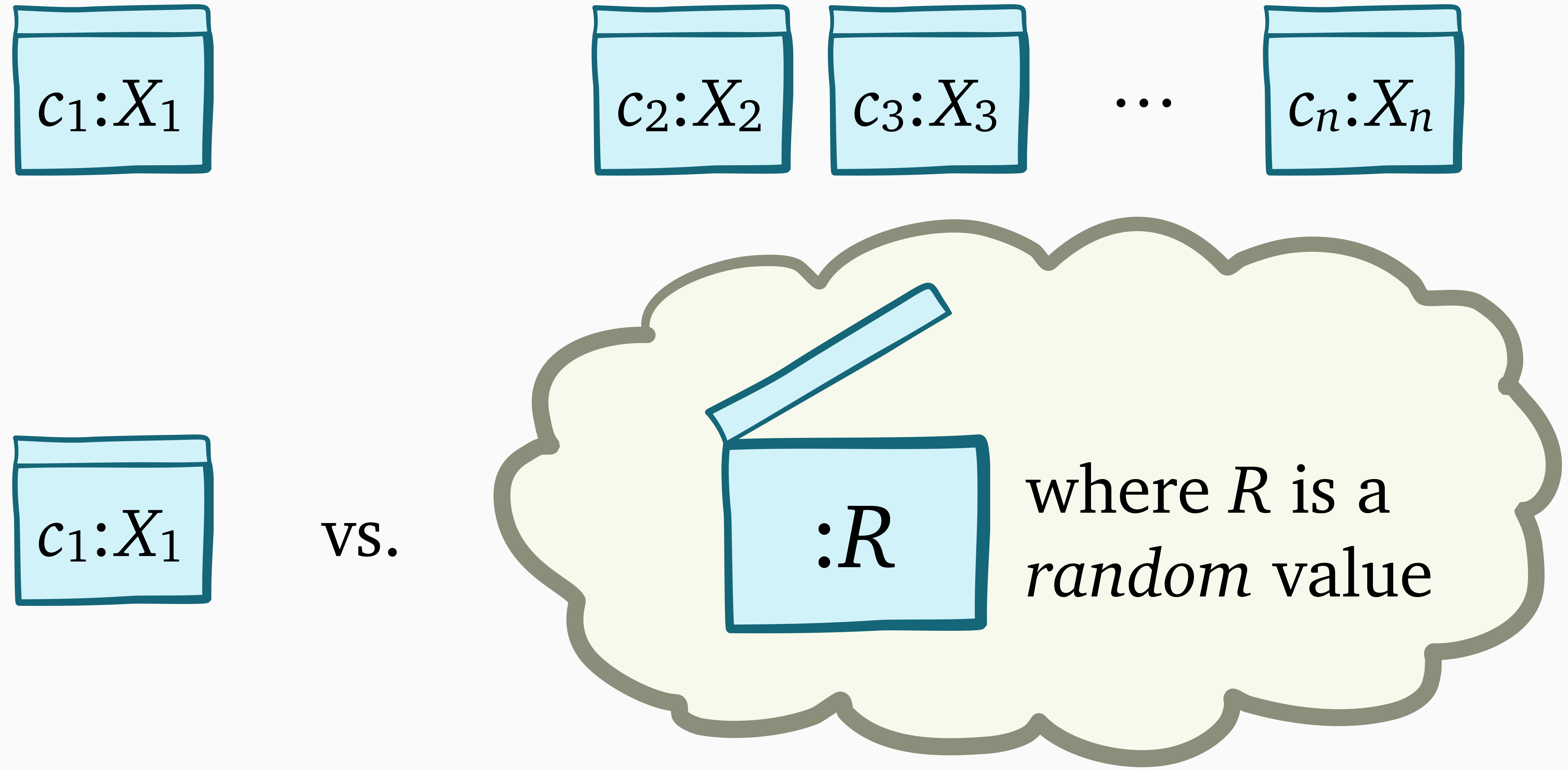
Why **Gittins** works under required inspection



Key property: $R \geq \min\{g(c_2{:}X_2), \ldots, g(c_n{:}X_n)\}$

$\Rightarrow$ If $g(c_1{:}X_1)$ minimal, then $g(c_1{:}X_1) \leq R$

Why **Gittins** works under required inspection

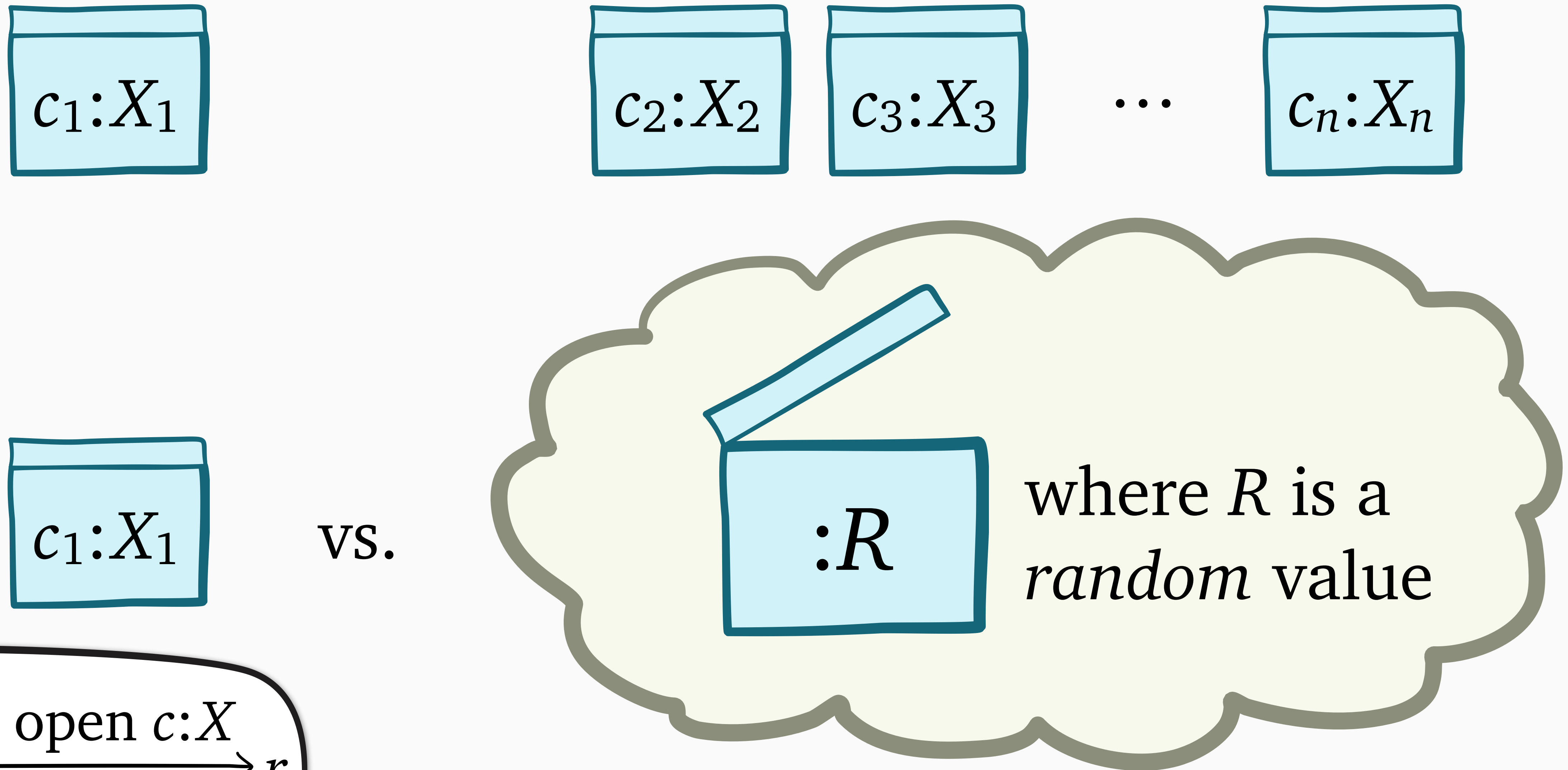


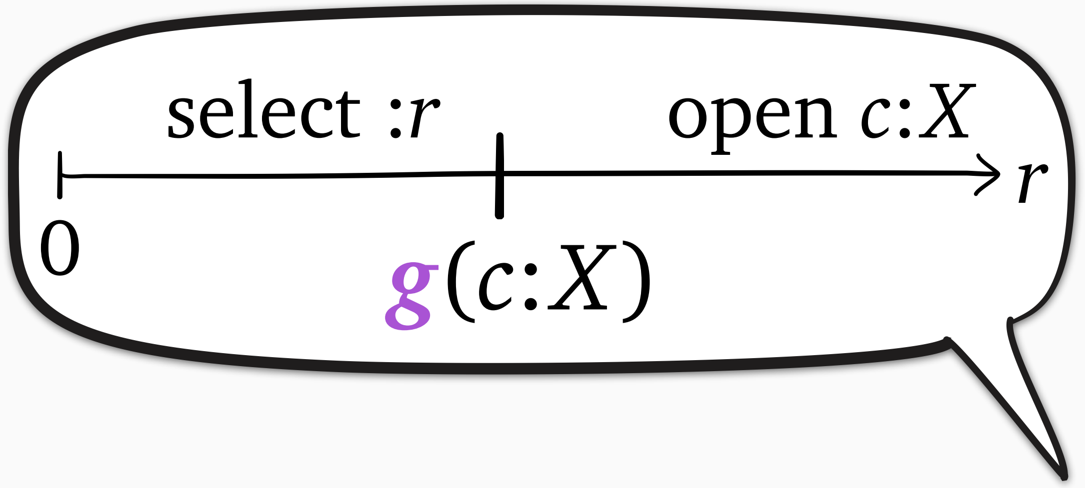
Key property: $R \geq \min\{g(c_2{:}X_2), \ldots, g(c_n{:}X_n)\}$

$\Rightarrow$ If $g(c_1{:}X_1)$ minimal, then $g(c_1{:}X_1) \leq R$

$\Rightarrow$ If $g(c_1{:}X_1)$ minimal, optimal to open box 1

Why Gittins works under required inspection





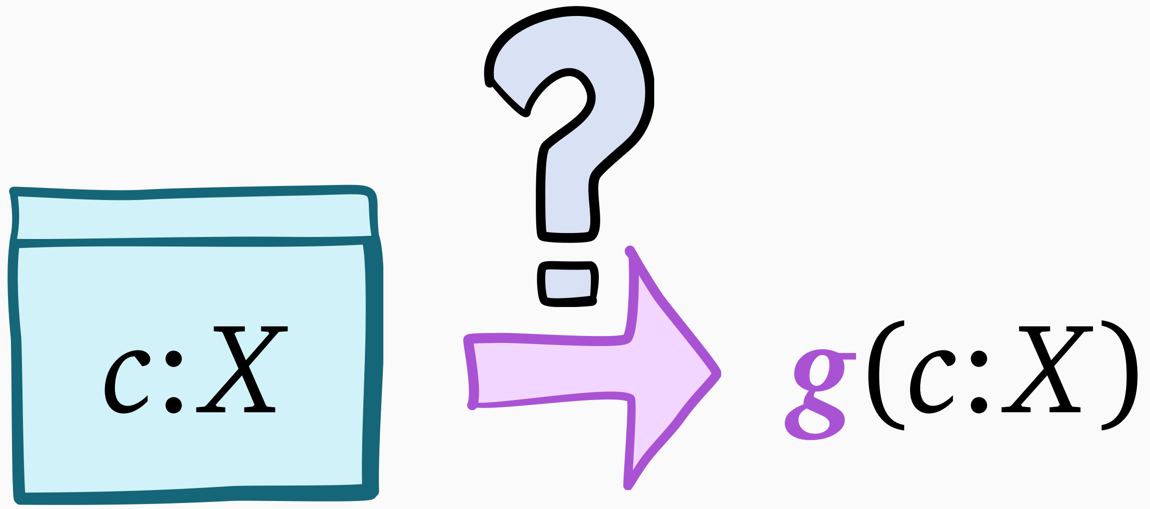
Key property: $R \geq \min\{g(c_2{:}X_2), \ldots, g(c_n{:}X_n)\}$

$\Rightarrow$ If $g(c_1{:}X_1)$ minimal, then $g(c_1{:}X_1) \leq R$

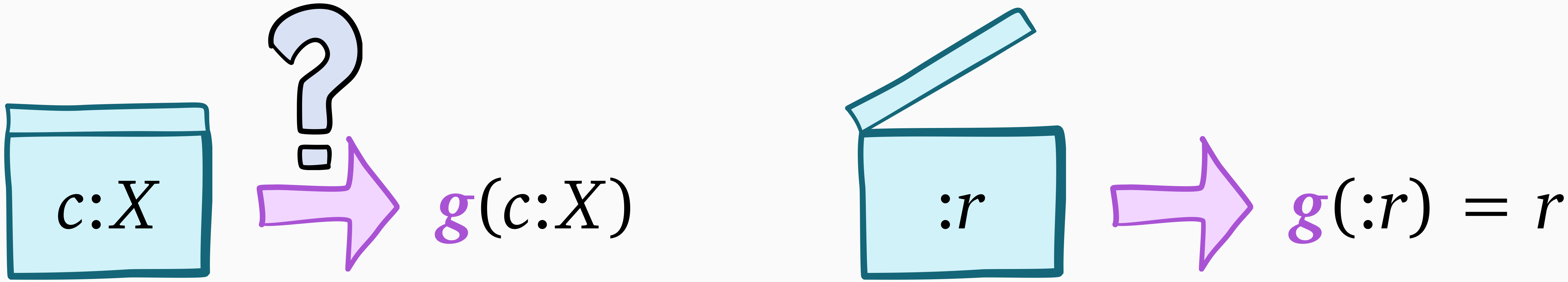
$\Rightarrow$ If $g(c_1{:}X_1)$ minimal, optimal to open box 1

What changes under optional inspection?

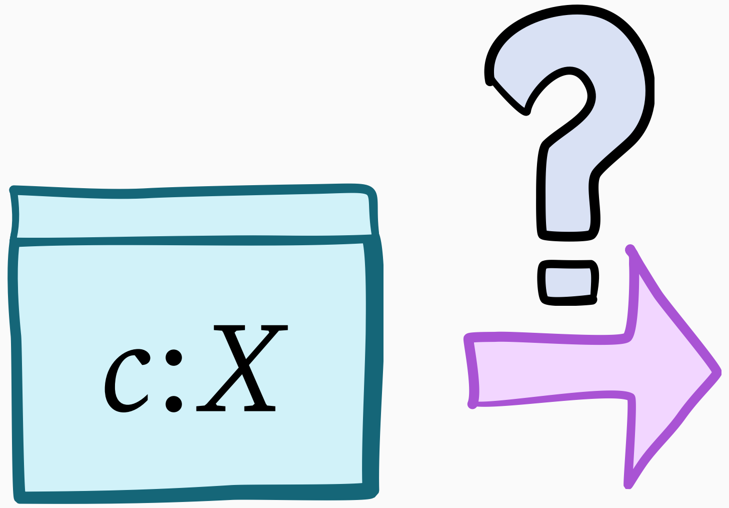
What changes under optional inspection?



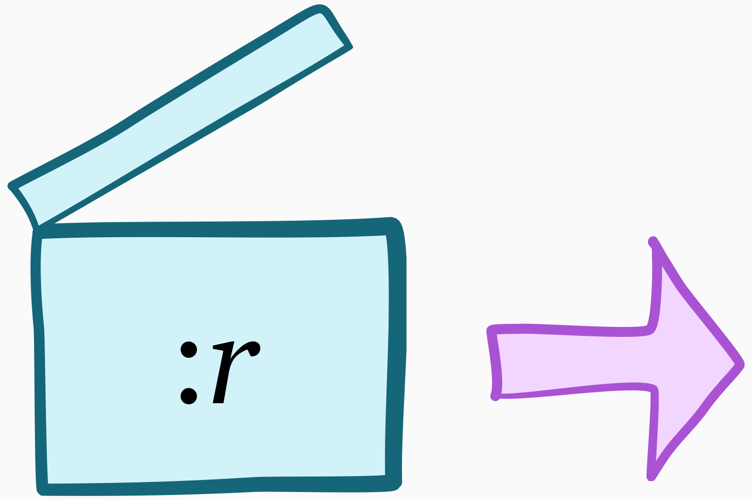
What changes under optional inspection?



What changes under optional inspection?

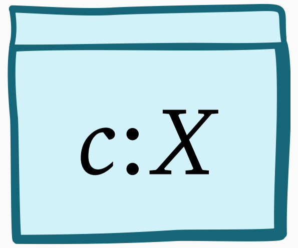


$g(c{:}X)$

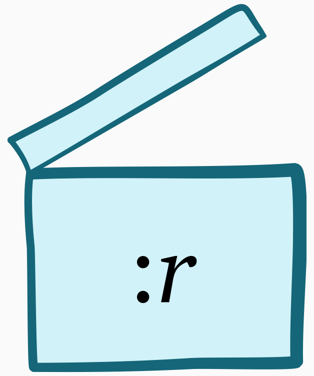


$g({:}r) = r$

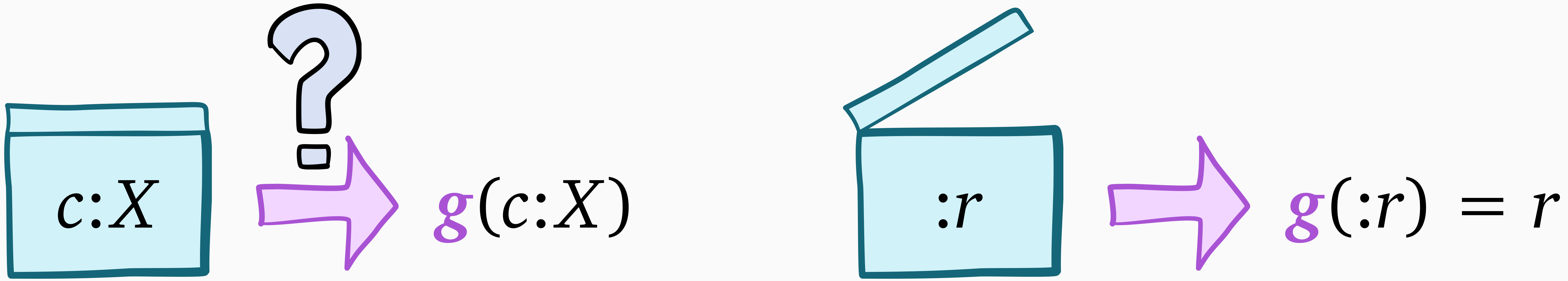
1.5-box problem



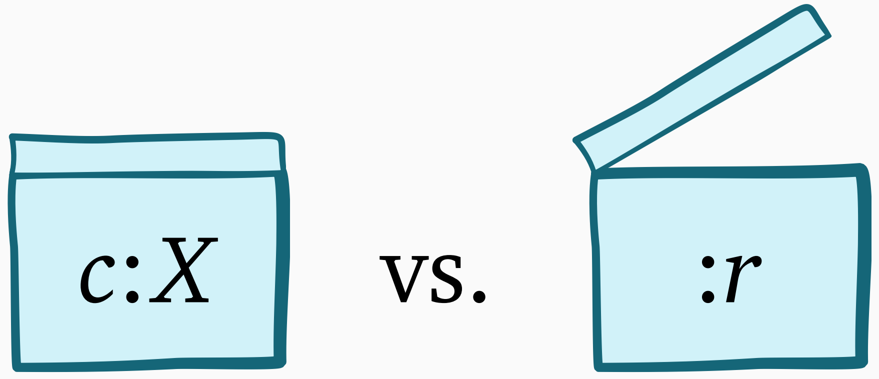
vs.



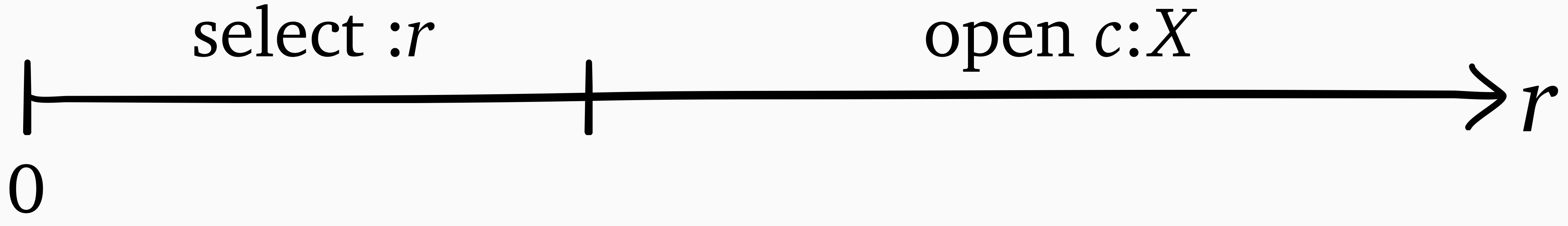
What changes under optional inspection?



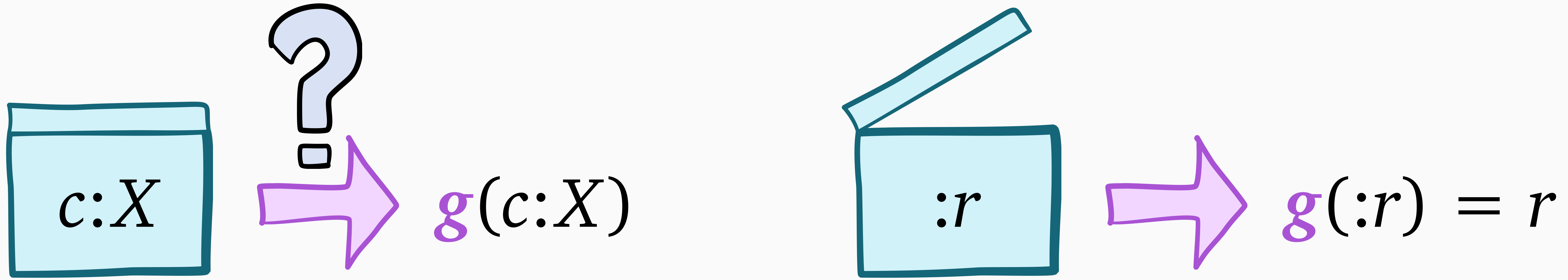
1.5-box problem



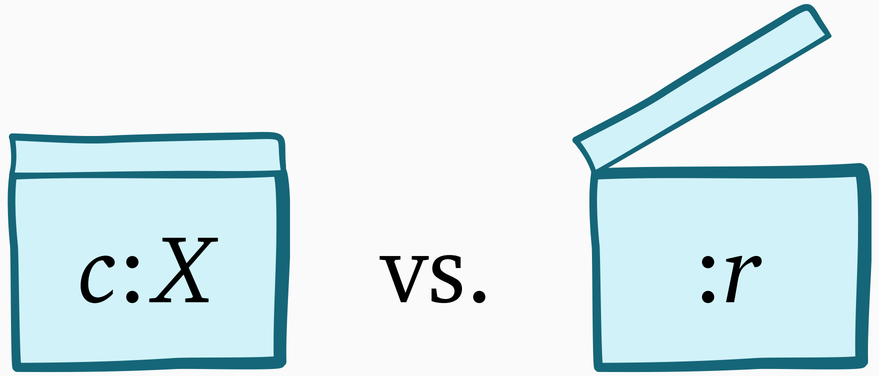
Key question: what to do in 1.5-box problem?



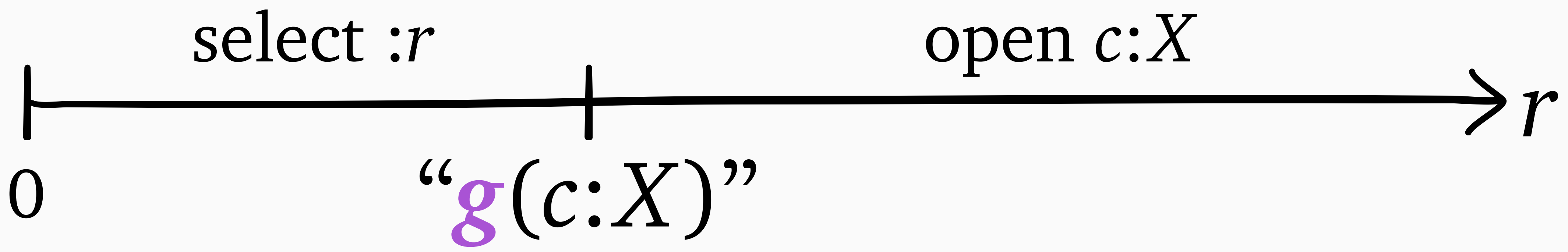
What changes under optional inspection?



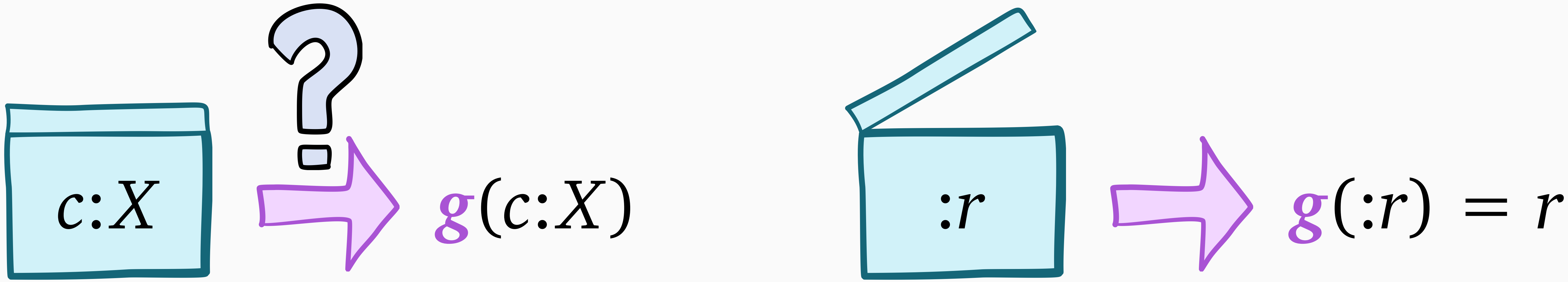
1.5-box problem



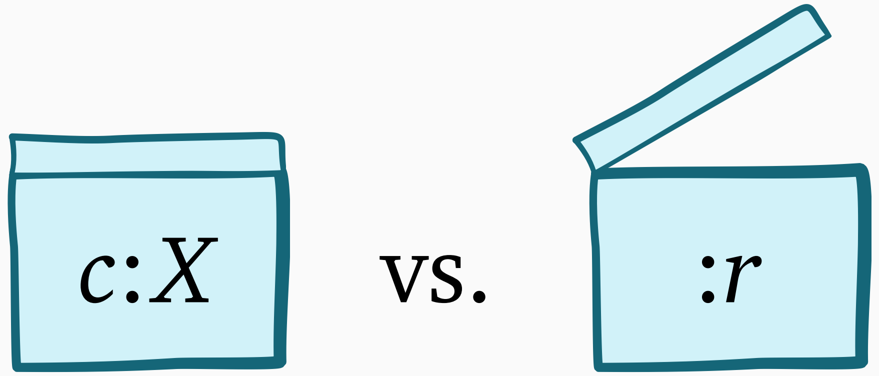
Key question: what to do in 1.5-box problem?



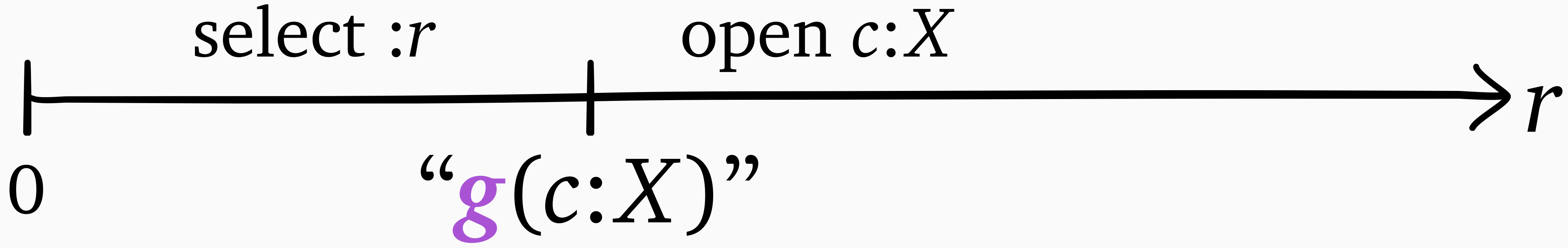
What changes under optional inspection?



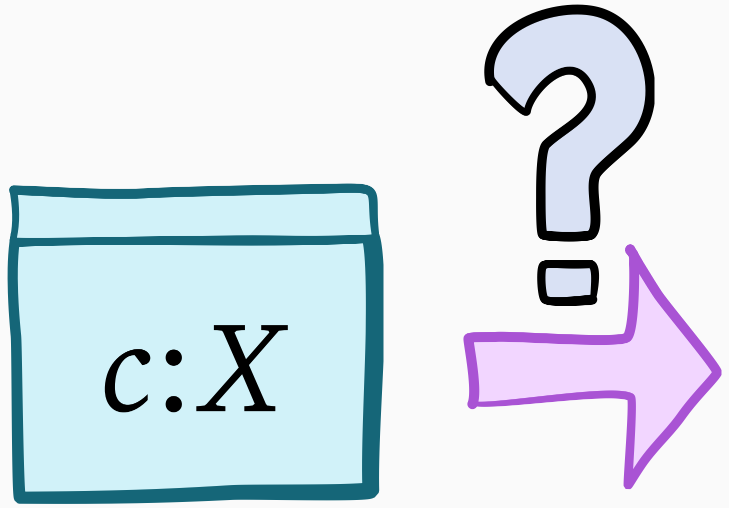
1.5-box problem



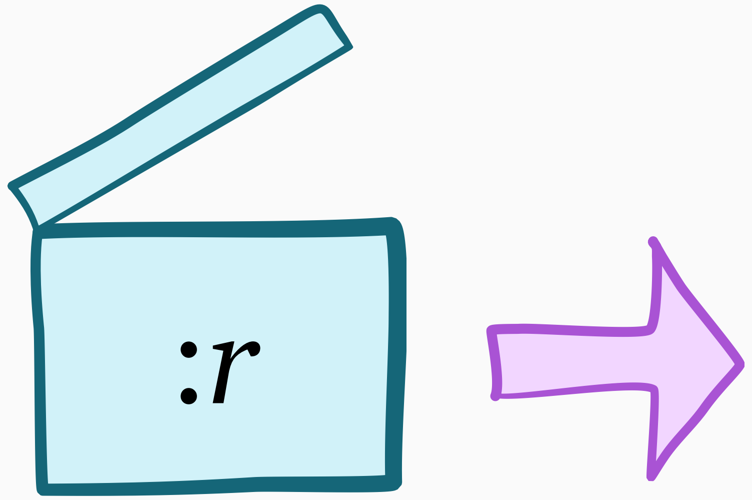
Key question: what to do in 1.5-box problem?



What changes under optional inspection?

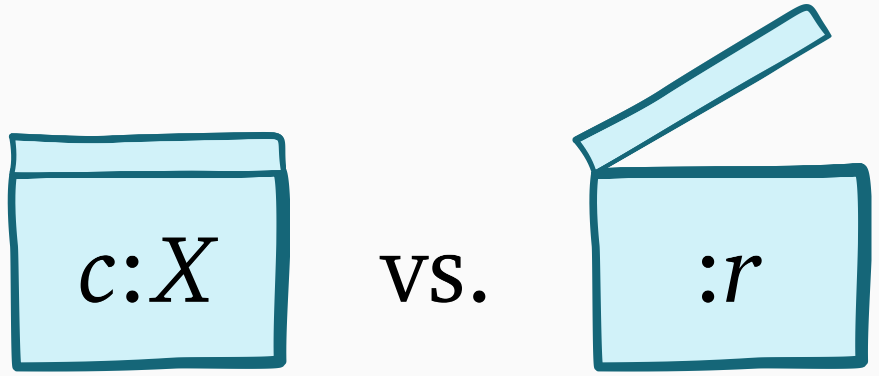


$g(c:X)$

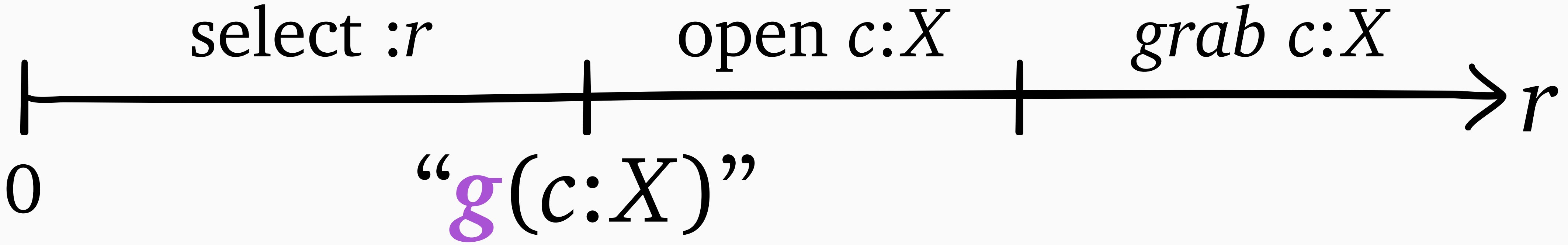


$g(:r) = r$

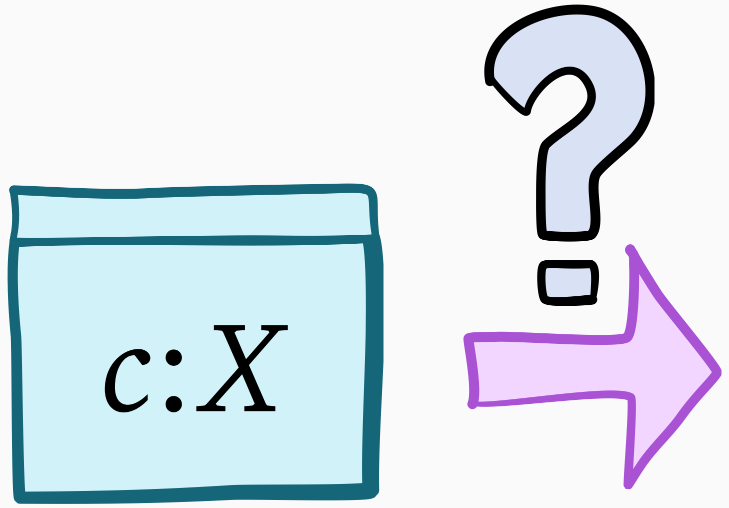
1.5-box problem



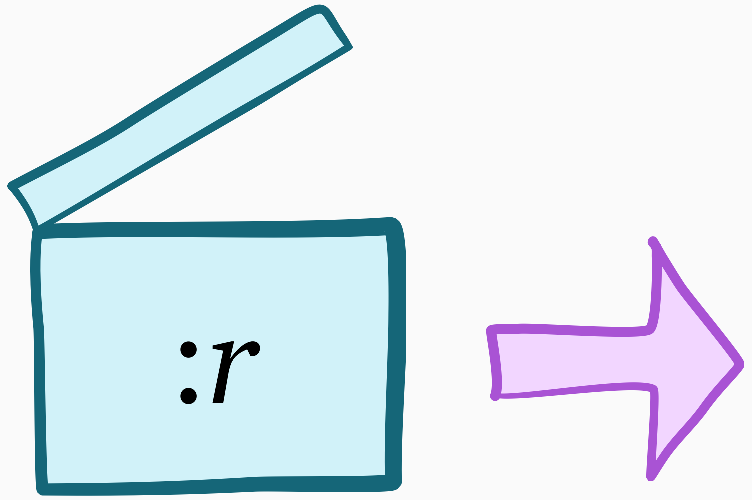
Key question: what to do in 1.5-box problem?



What changes under optional inspection?

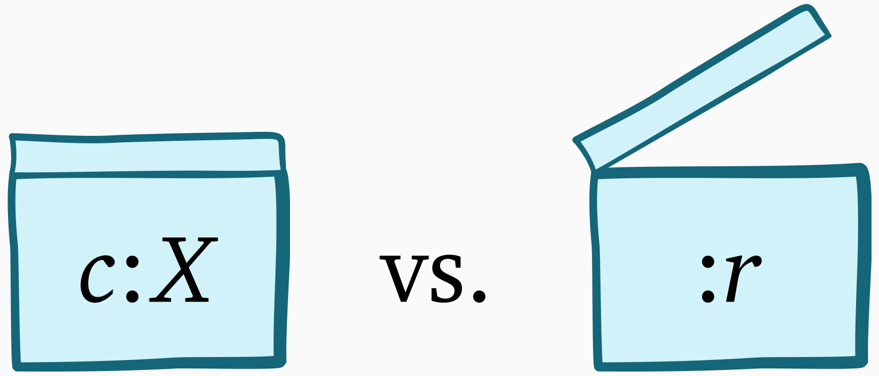


$g(c{:}X)$

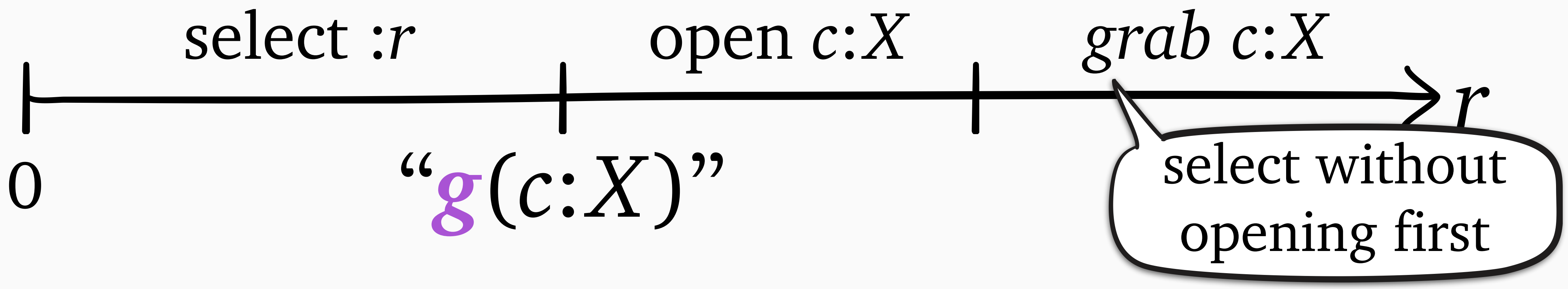


$g({:}r) = r$

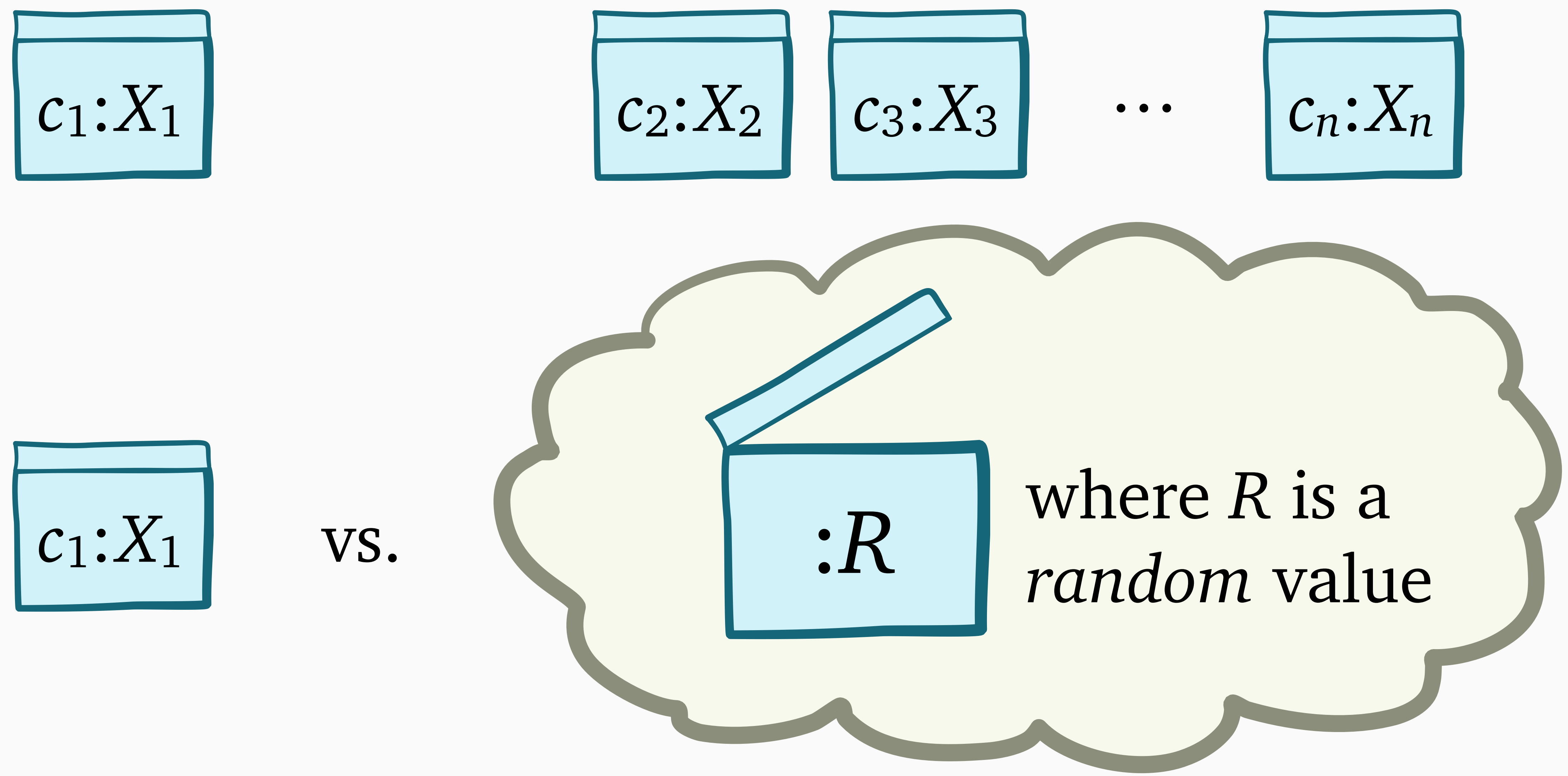
1.5-box problem



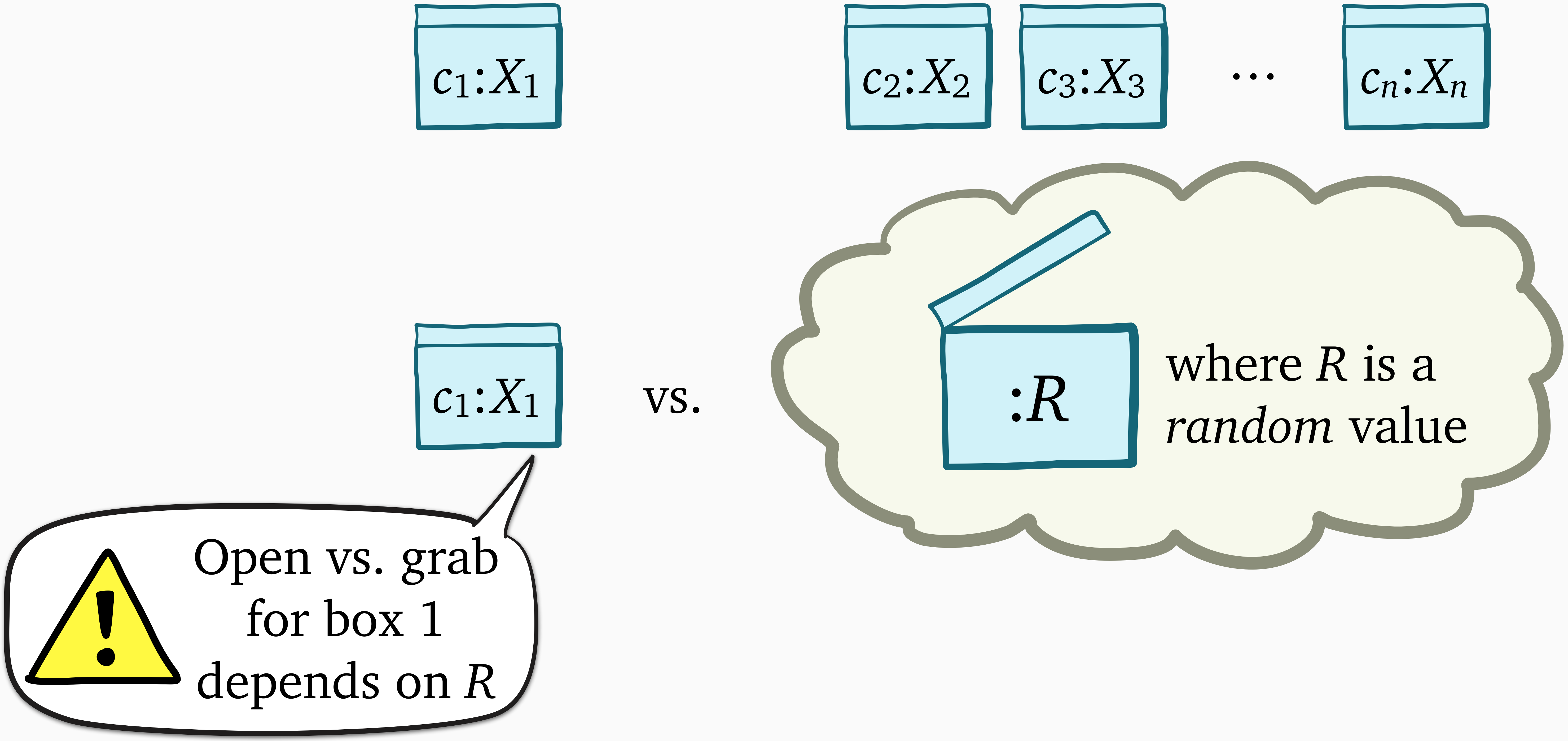
Key question: what to do in 1.5-box problem?



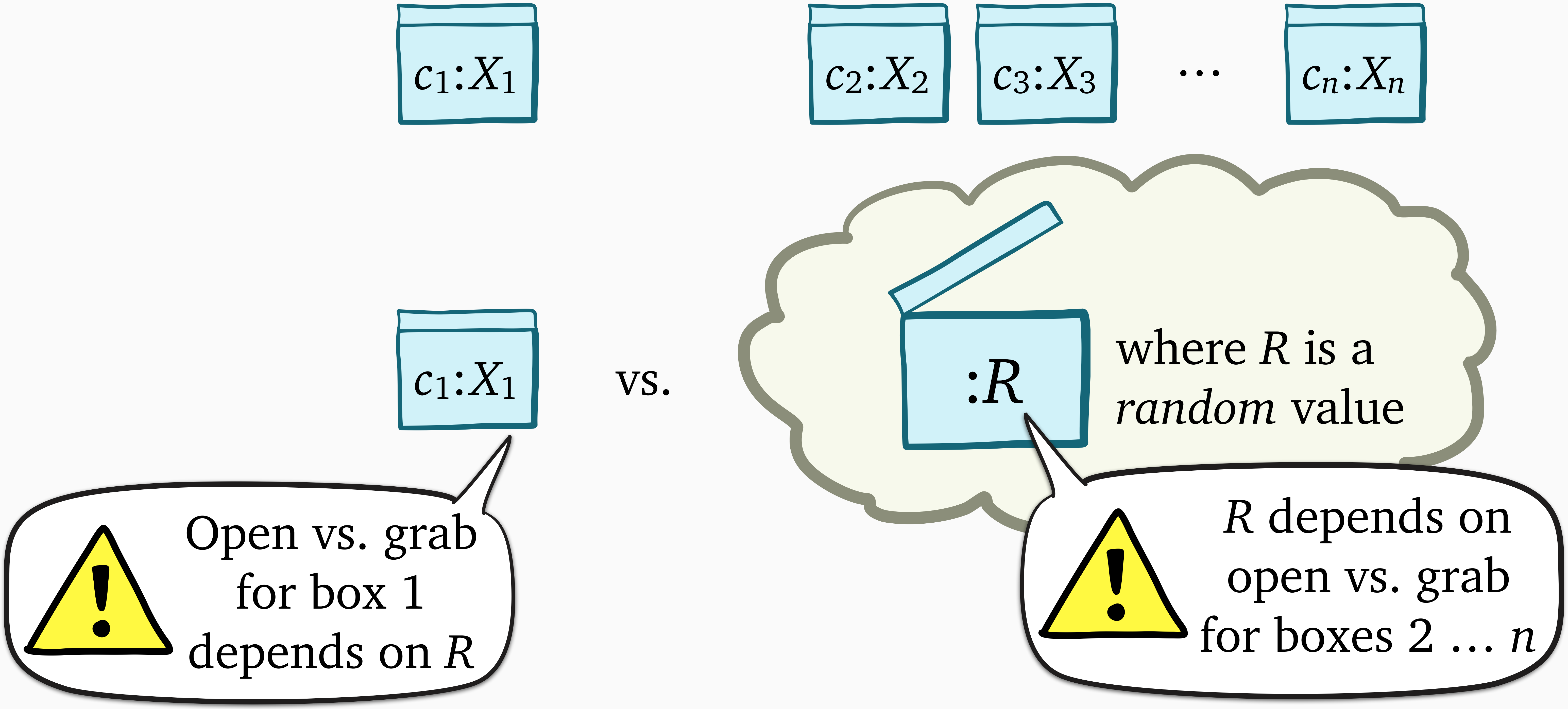
What goes wrong under optional inspection?



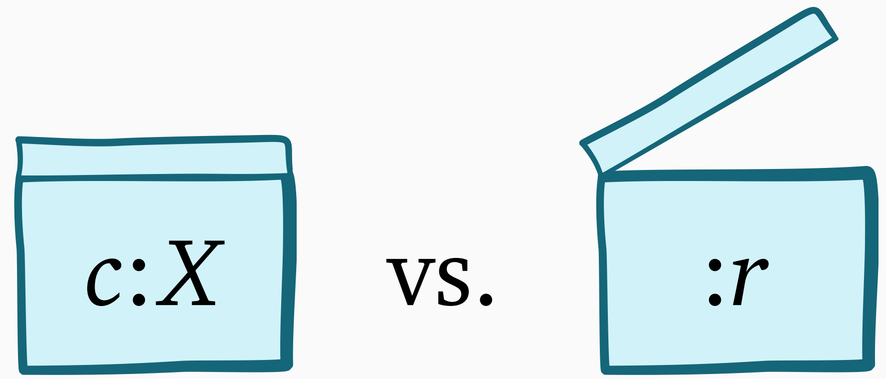
What goes wrong under optional inspection?

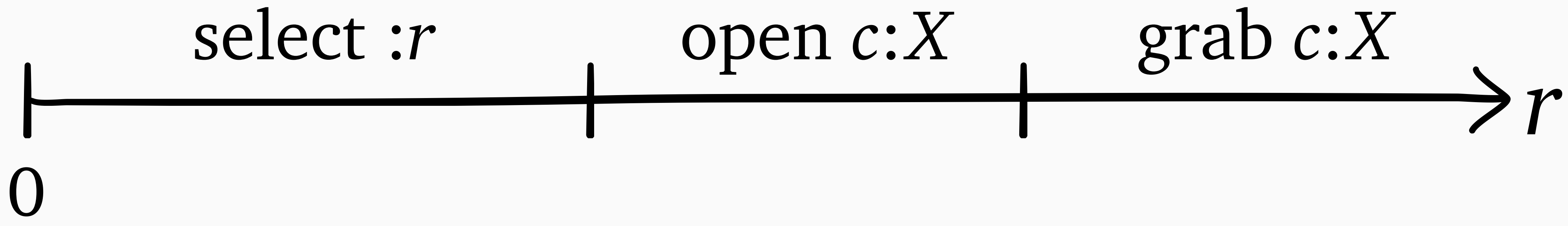


What goes wrong under optional inspection?

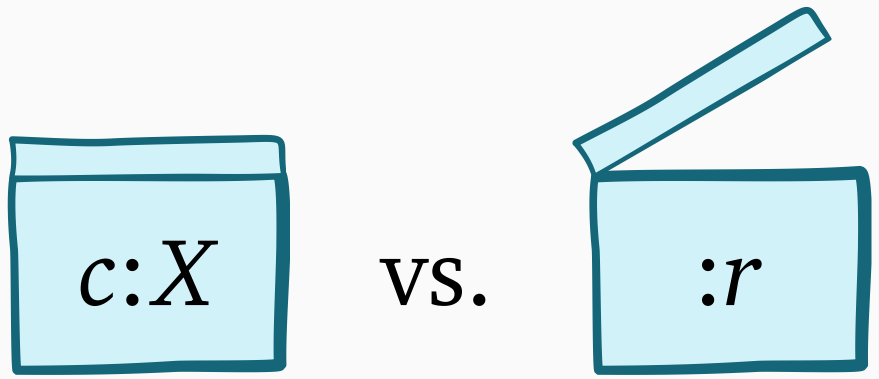


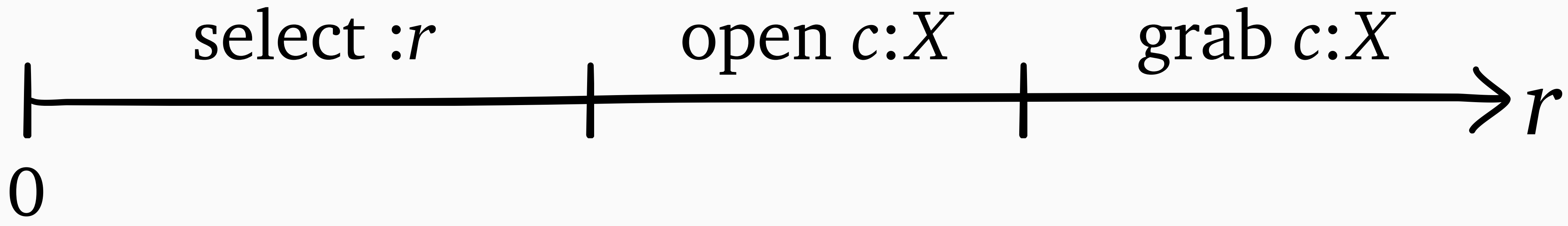
Approximate solution with **Local Hedging**

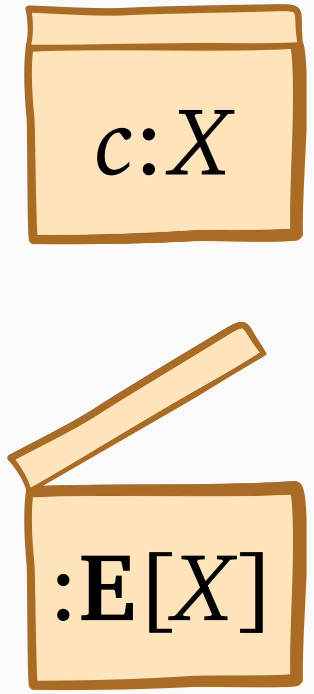




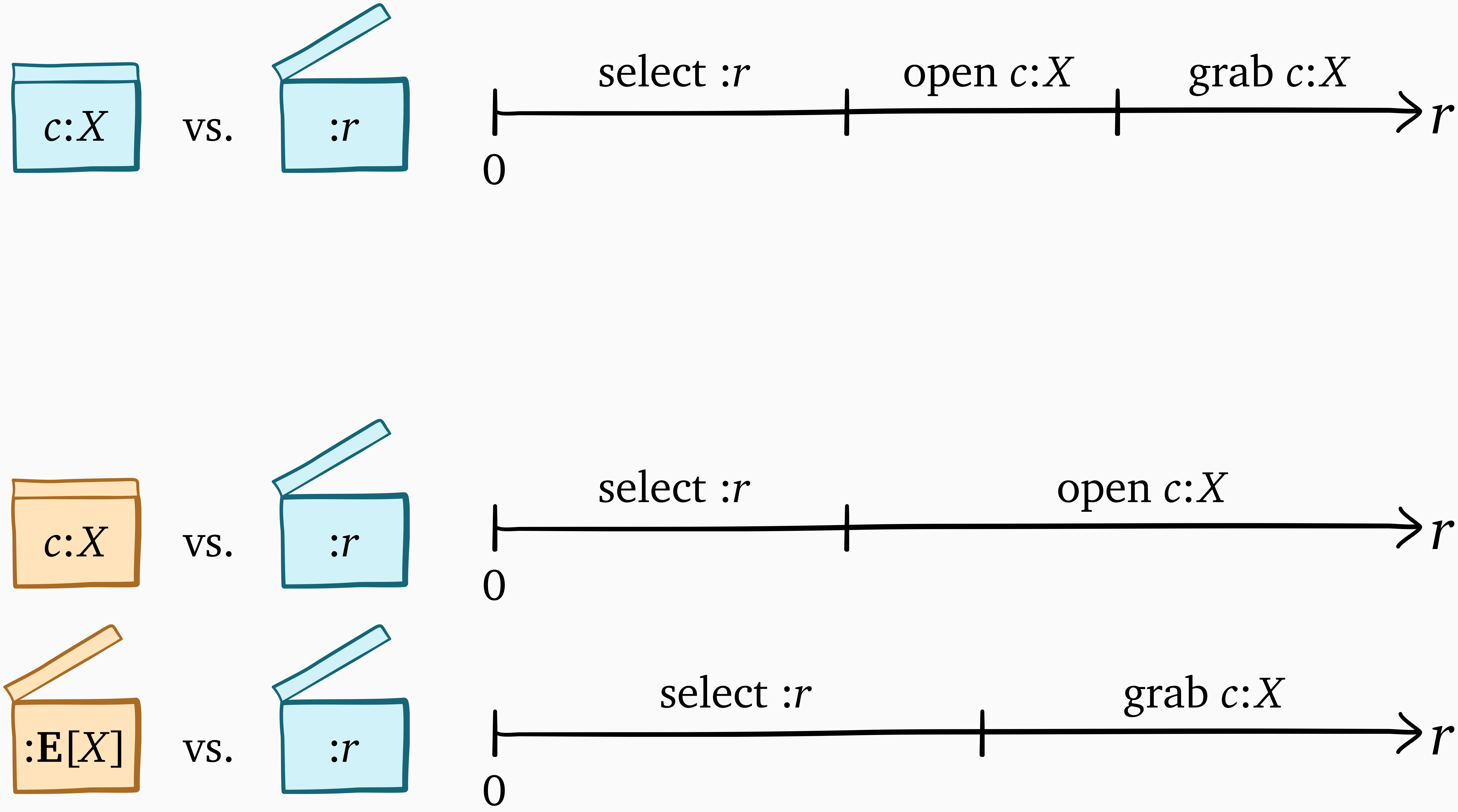
Approximate solution with **Local Hedging**



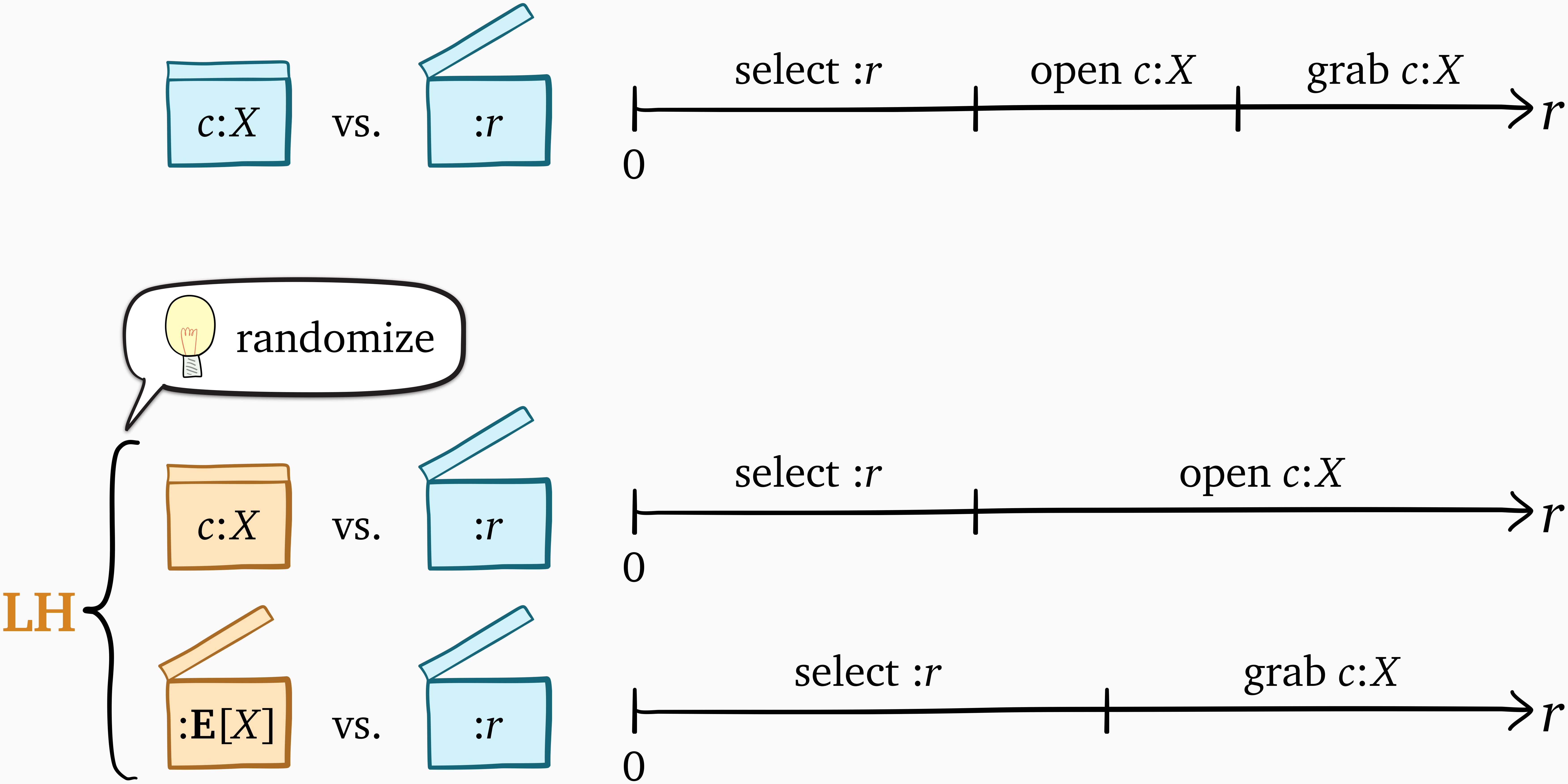




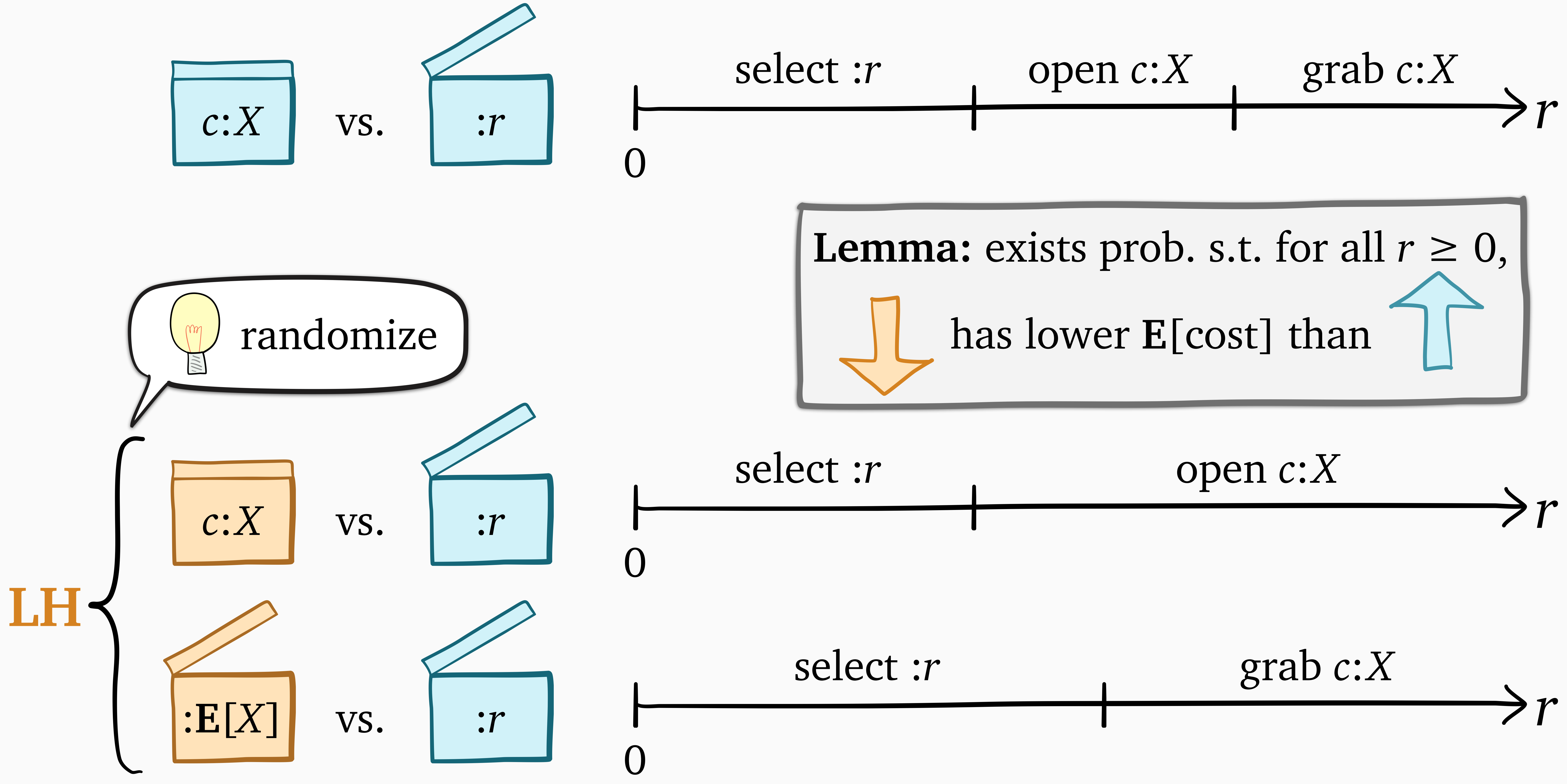
Approximate solution with **Local Hedging**



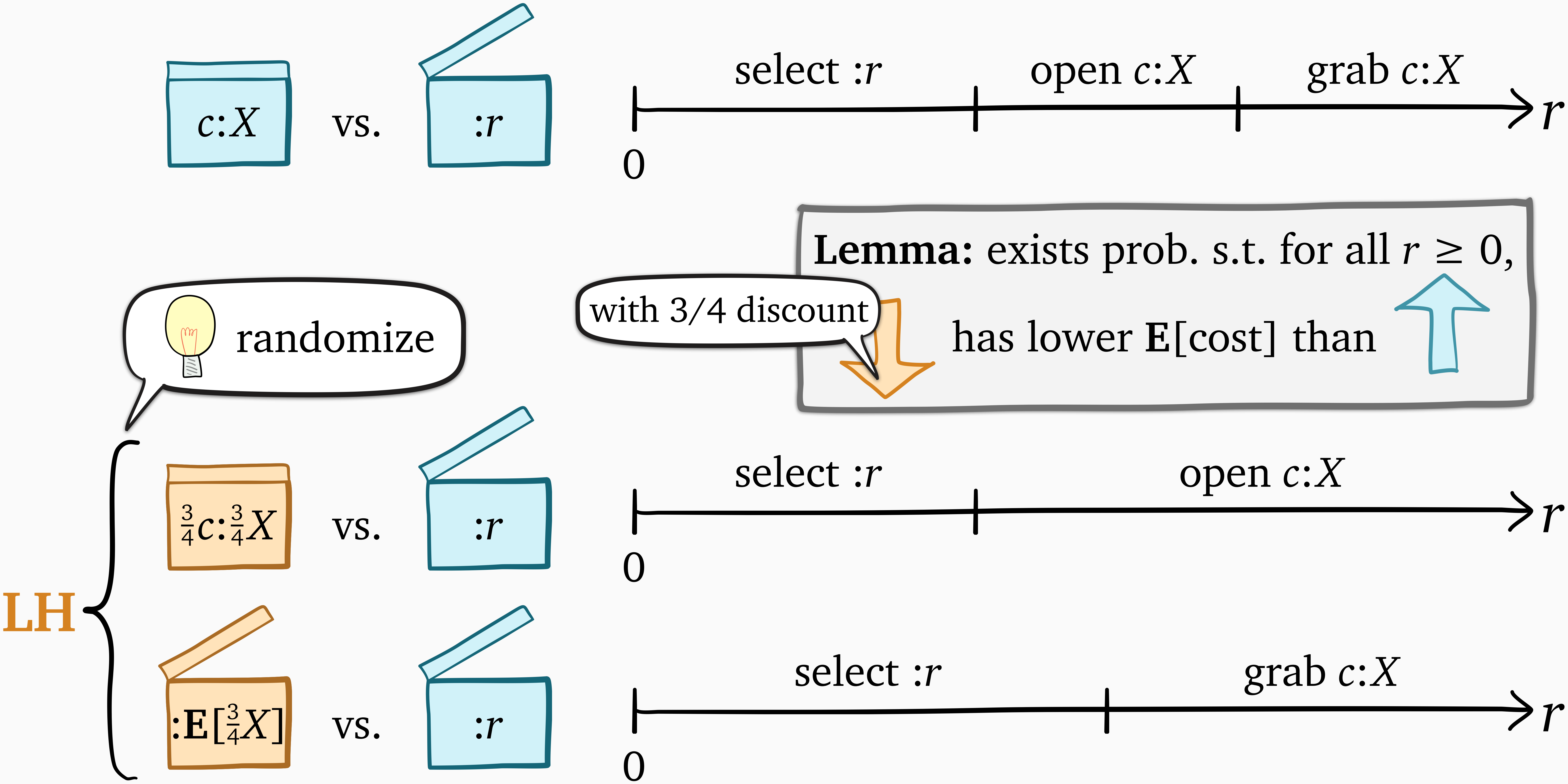
Approximate solution with **Local Hedging**



Approximate solution with **Local Hedging**



Approximate solution with **Local Hedging**



Approximate solution with **Local Hedging**

$c{:}X$ vs. $:r$ $\qquad \mathbf{E}[\text{cost}] = ?$

Lemma: exists prob. s.t. for all $r \geq 0$,

with 3/4 discount ↓ has lower $\mathbf{E}[\text{cost}]$ than ↑

randomize

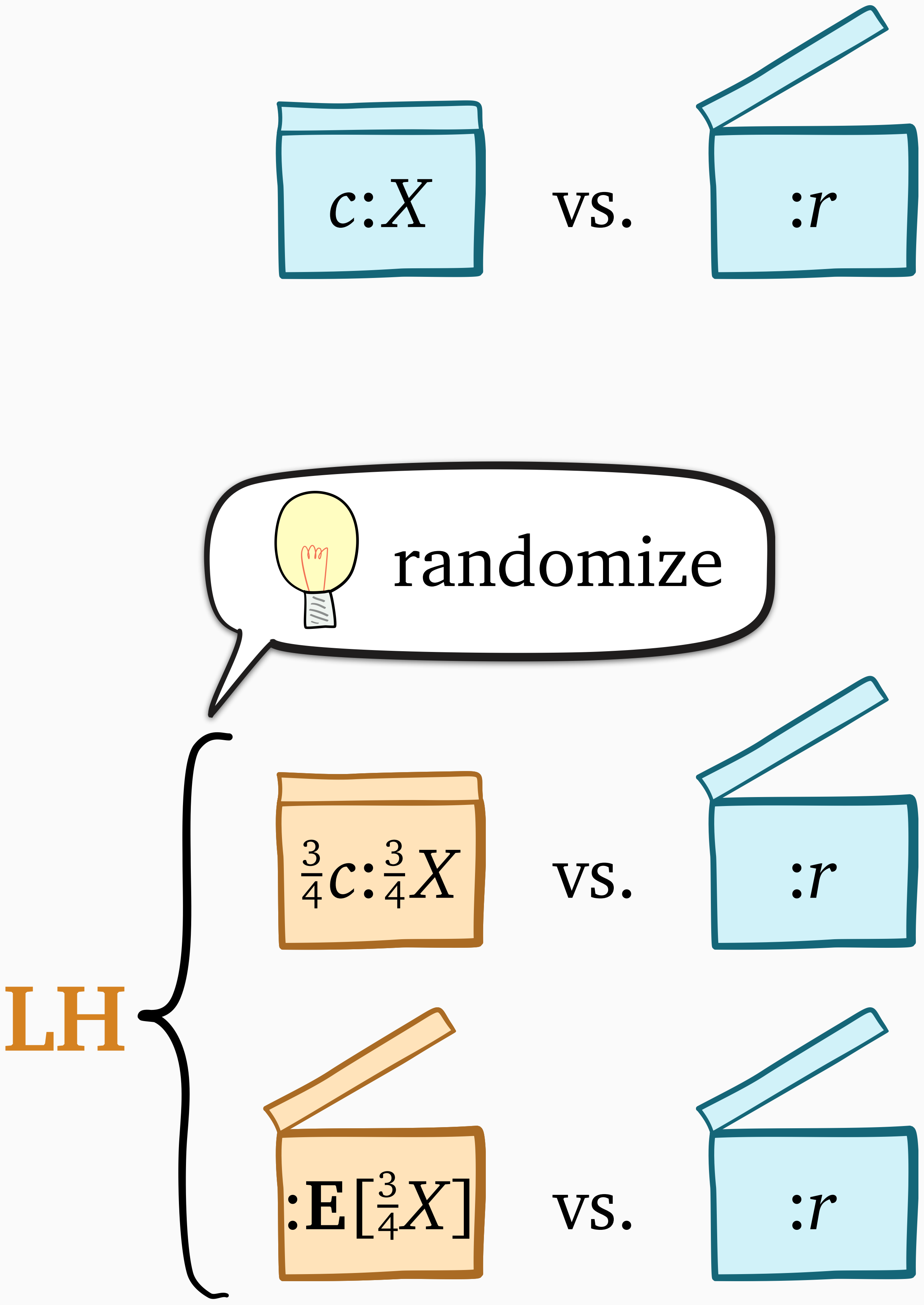
LH

$\frac{3}{4}c{:}\frac{3}{4}X$ vs. $:r$

$:\mathbf{E}[\frac{3}{4}X]$ vs. $:r$

$\mathbf{E}[\text{cost}] = ?$

Approximate solution with **Local Hedging**



$$\mathbf{E}[\text{cost}] = \min\Big(r,\ \mathbf{E}[X],\ c + \mathbf{E}[\min(r, X)]\Big)$$

Lemma: exists prob. s.t. for all $r \geq 0$,

with 3/4 discount

has lower $\mathbf{E}[\text{cost}]$ than

$$\mathbf{E}[\text{cost}] = ?$$

Approximate solution with **Local Hedging**

$c{:}X$ vs. $:r$

$$\mathbf{E}[\text{cost}] = \min\Big(r,\ \mathbf{E}[X],\ c + \mathbf{E}[\min(r, X)]\Big)$$

randomize

Lemma: exists prob. s.t. for all $r \geq 0$,

with 3/4 discount

has lower $\mathbf{E}[\text{cost}]$ than

LH

$\tfrac{3}{4}c{:}\tfrac{3}{4}X$ vs. $:r$

$$\mathbf{E}[\text{cost} \mid \text{open}] = \min\Big(r,\ \tfrac{3}{4}c + \mathbf{E}\big[\min(r, \tfrac{3}{4}X)\big]\Big)$$

$:\mathbf{E}[\tfrac{3}{4}X]$ vs. $:r$

$$\mathbf{E}[\text{cost}] = \ ?$$

Approximate solution with **Local Hedging**

$c{:}X$ vs. $:r$

$$\mathbf{E}[\text{cost}] = \min\Big(r,\ \mathbf{E}[X],\ c + \mathbf{E}[\min(r, X)]\Big)$$

randomize

Lemma: exists prob. s.t. for all $r \geq 0$,

with 3/4 discount

has lower $\mathbf{E}[\text{cost}]$ than

LH

$\frac{3}{4}c{:}\frac{3}{4}X$ vs. $:r$

$$\mathbf{E}[\text{cost} \mid \text{open}] = \min\Big(r,\ \tfrac{3}{4}c + \mathbf{E}\big[\min(r, \tfrac{3}{4}X)\big]\Big)$$

$$\mathbf{E}[\text{cost}] = ?$$

$:\mathbf{E}[\frac{3}{4}X]$ vs. $:r$

$$\mathbf{E}[\text{cost} \mid \text{grab}] = \min\Big(r,\ \mathbf{E}\big[\tfrac{3}{4}X\big]\Big)$$

Approximate solution with **Local Hedging**

$c{:}X$ vs. $:r$

$$\mathbf{E}[\text{cost}] = \min\Big(r,\ \mathbf{E}[X],\ c + \mathbf{E}[\min(r, X)]\Big)$$

Lemma: exists prob. s.t. for all $r \geq 0$,

with 3/4 discount ⇓ has lower $\mathbf{E}[\text{cost}]$ than ⇑

randomize

LH

$\frac{3}{4}c{:}\frac{3}{4}X$ vs. $:r$

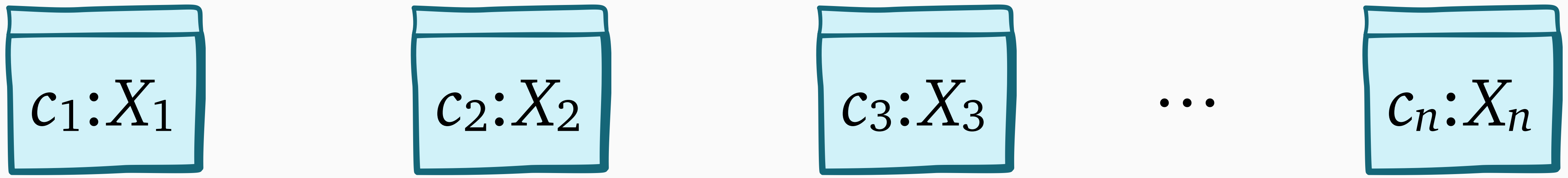
$$\mathbf{E}[\text{cost} \mid \text{open}] = \min\Big(r,\ \tfrac{3}{4}c + \mathbf{E}\big[\min(r, \tfrac{3}{4}X)\big]\Big)$$

$:\mathbf{E}[\frac{3}{4}X]$ vs. $:r$

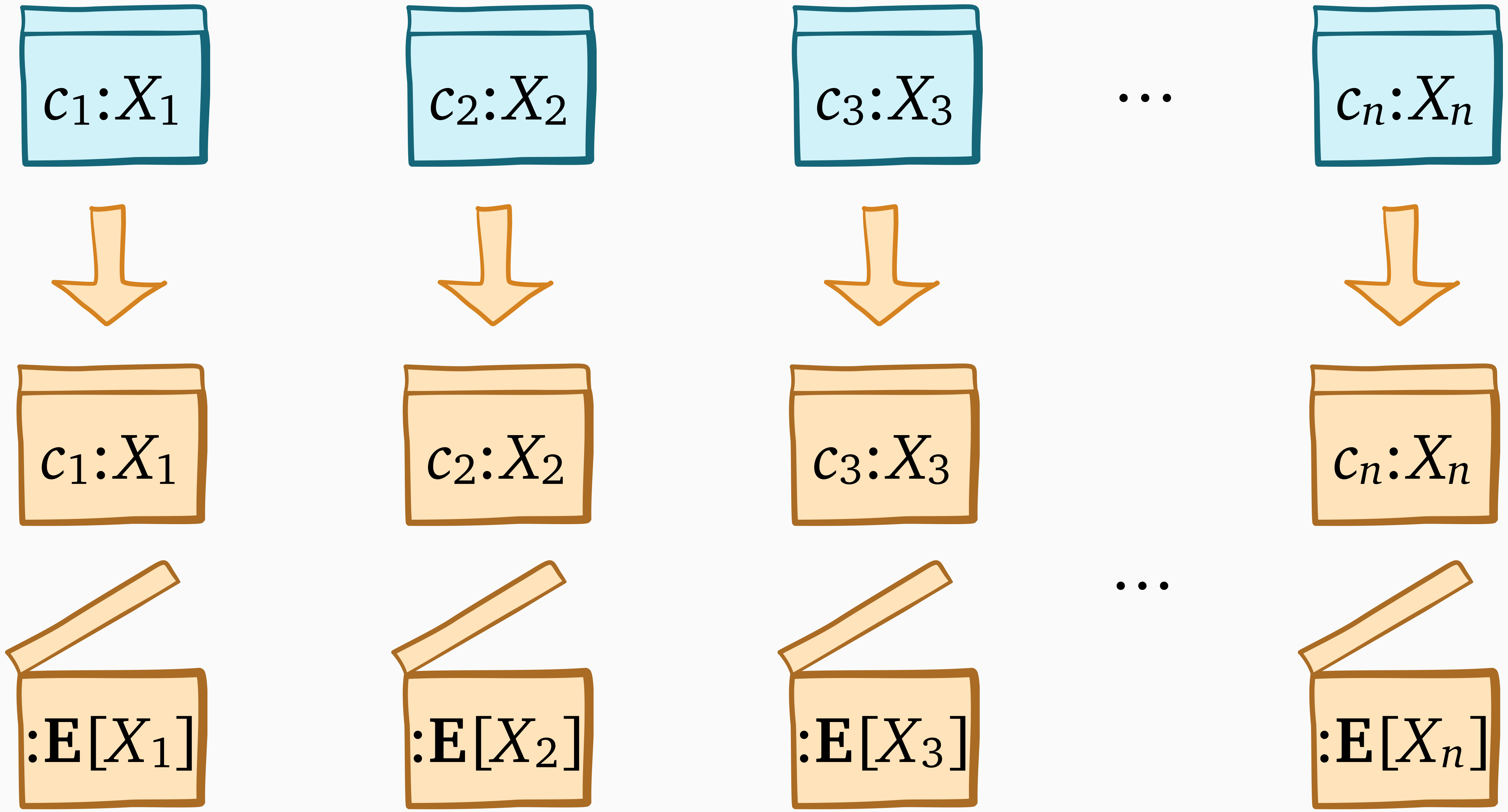
$$\mathbf{E}[\text{cost} \mid \text{grab}] = \min\Big(r,\ \mathbf{E}\big[\tfrac{3}{4}X\big]\Big)$$

$$\mathbf{E}[\text{cost}] = p\mathbf{E}[\text{cost} \mid \text{open}] + (1-p)\mathbf{E}[\text{cost} \mid \text{grab}]$$

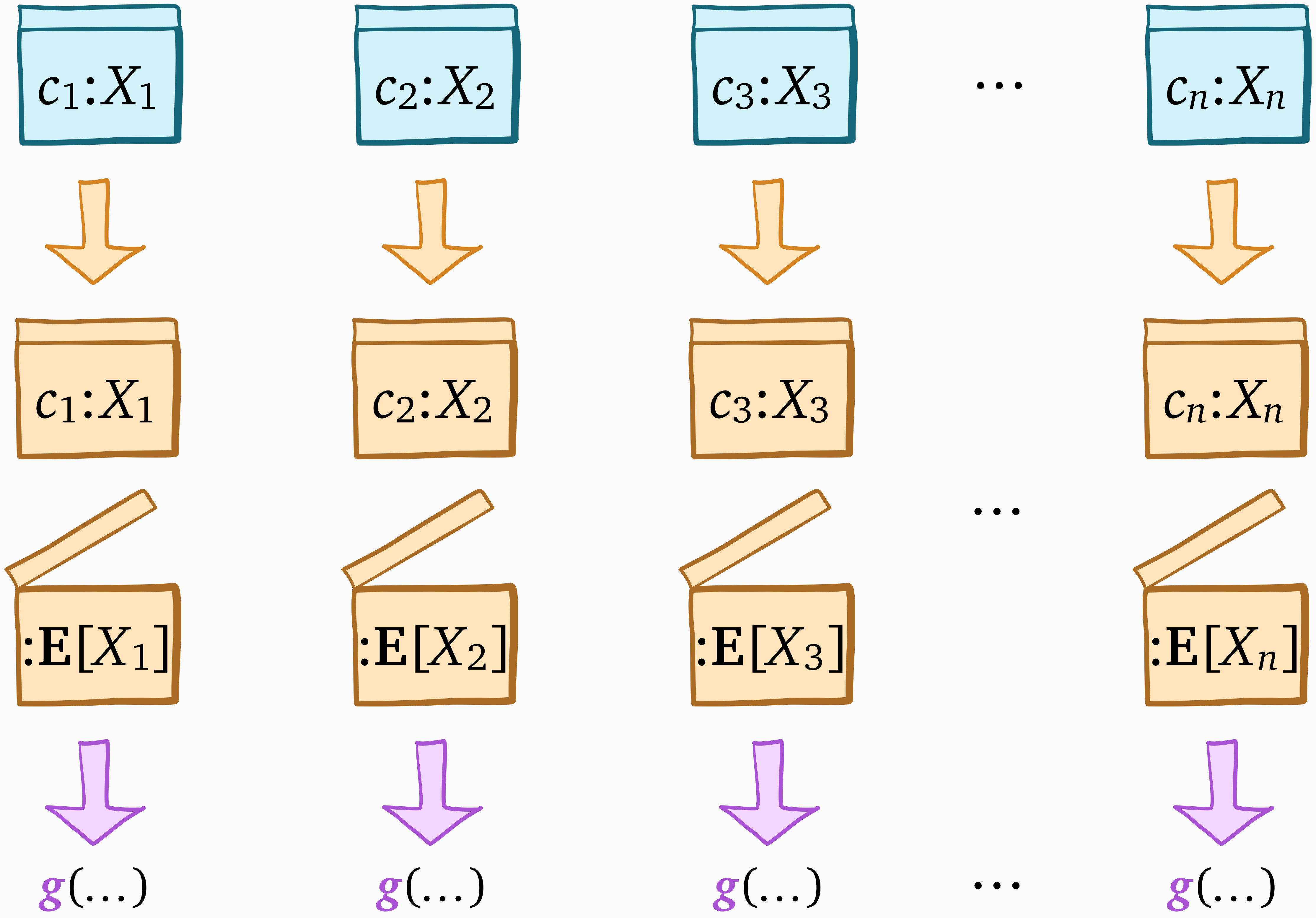
How **Local Hedging** helps



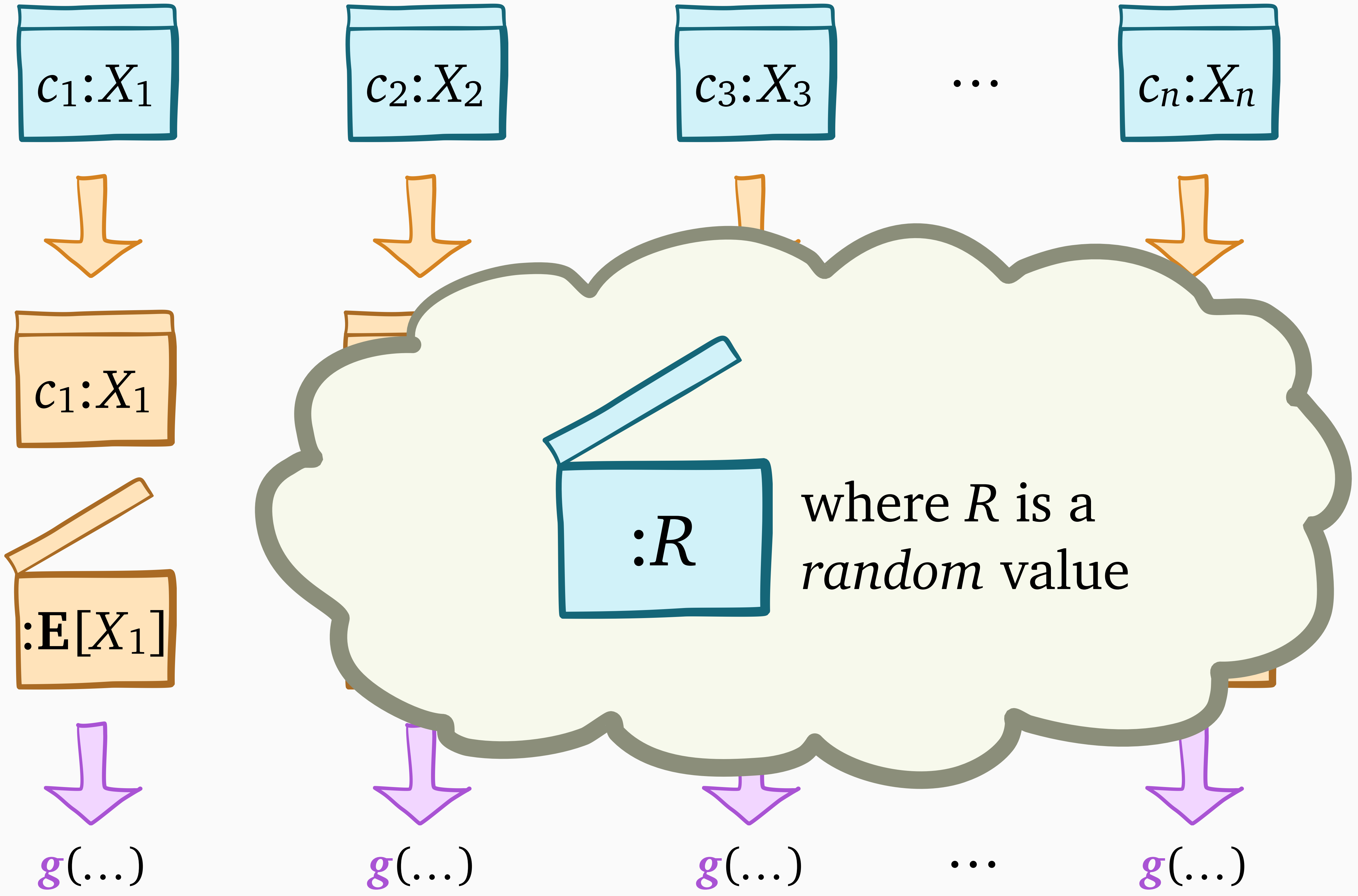
How **Local Hedging** helps



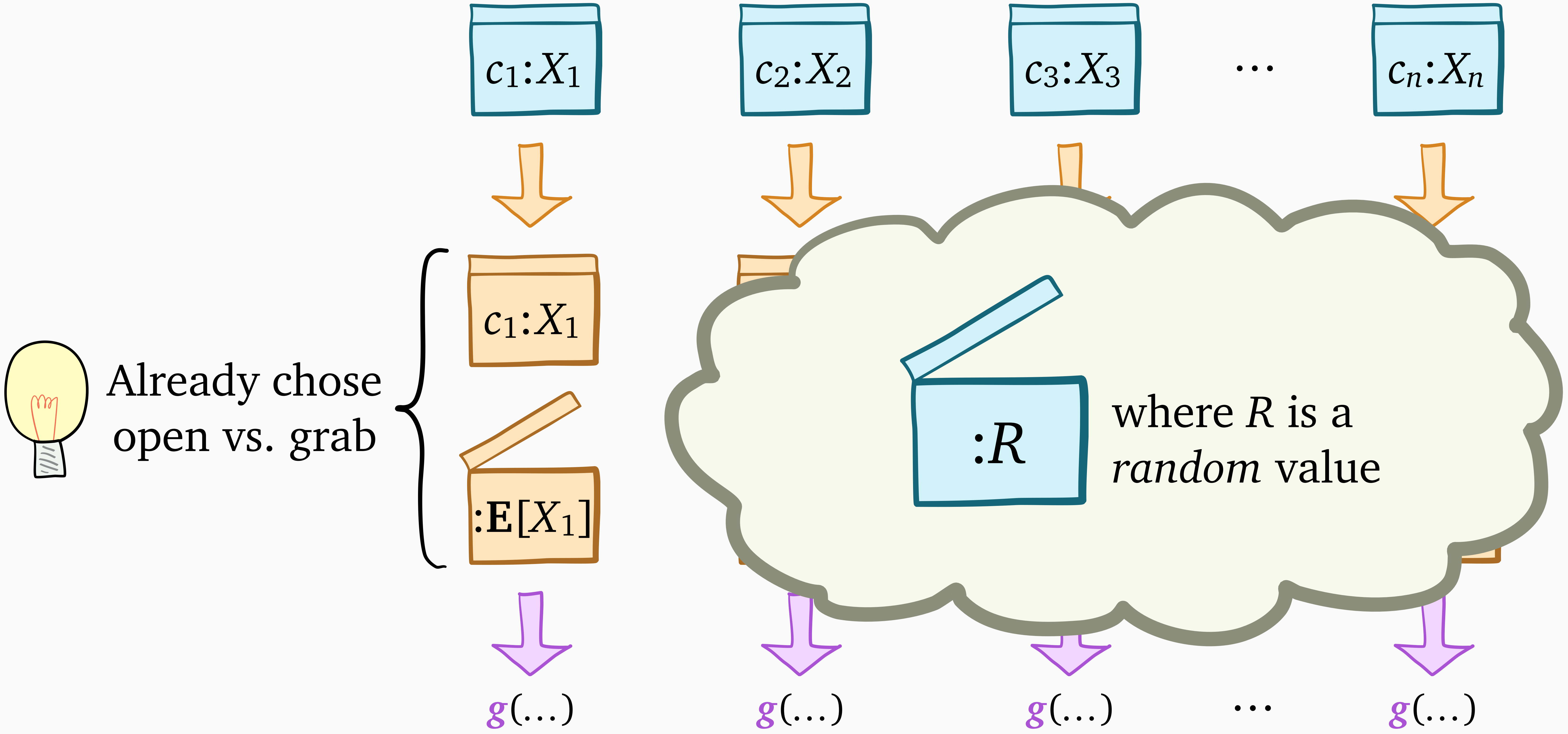
How **Local Hedging** helps



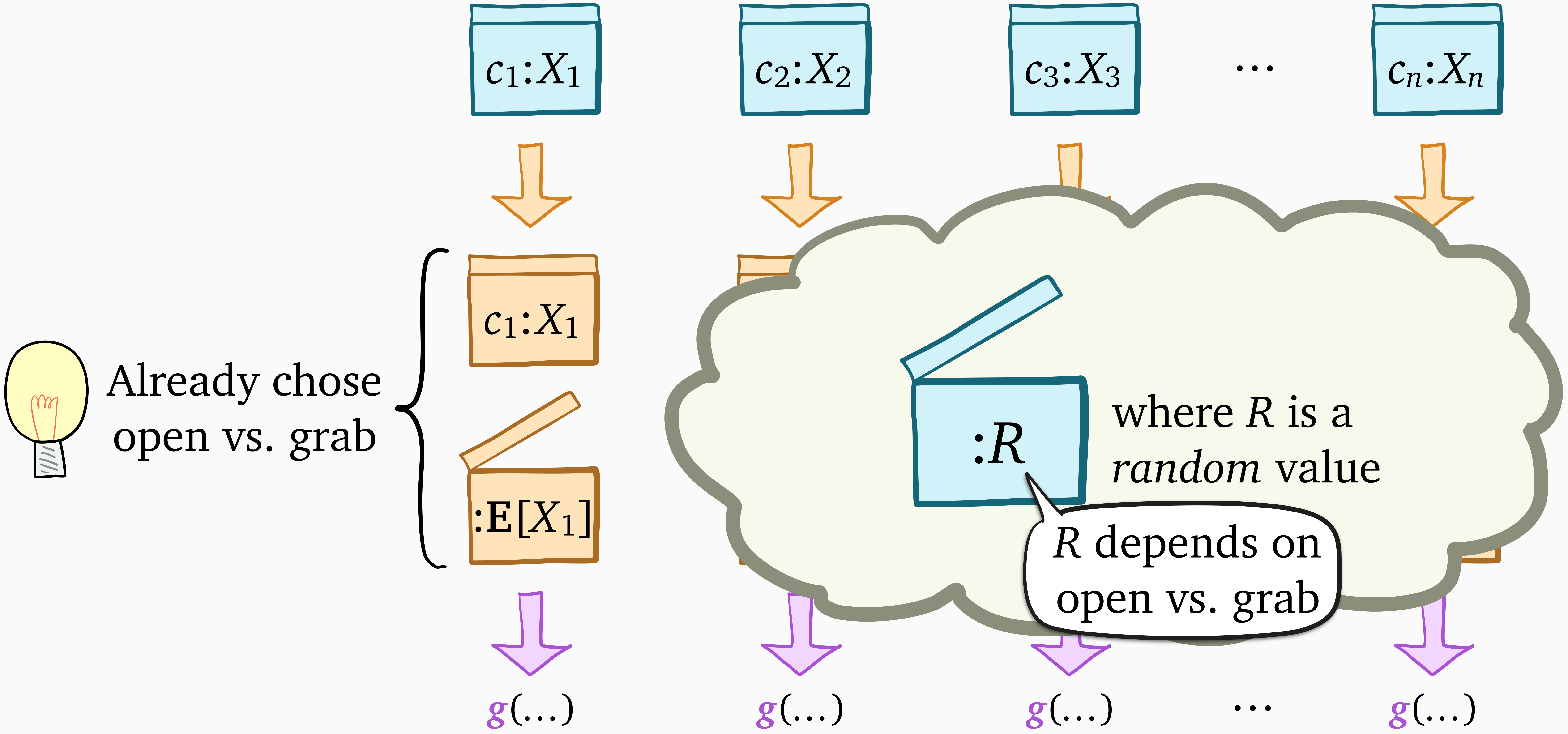
How **Local Hedging** helps



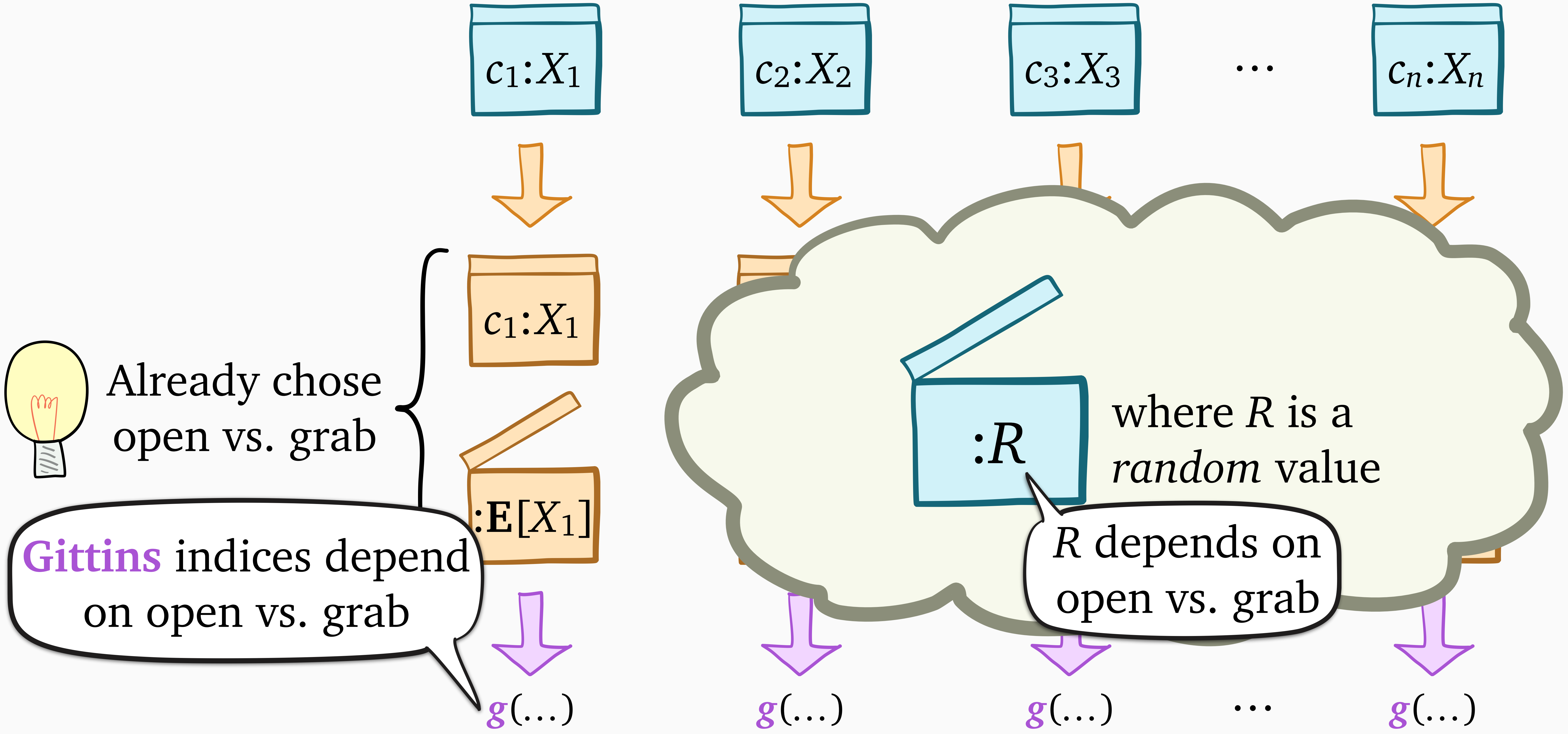
How Local Hedging helps



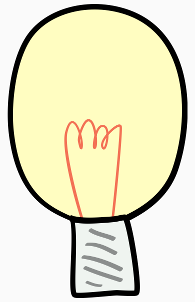
How **Local Hedging** helps



How **Local Hedging** helps



Our contribution



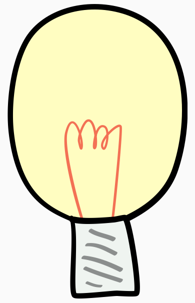
Local Hedging (LH)

New *decomposition-based* technique for optional inspection

- Reduces problem to required-inspection case
- Naturally generalizes to combinatorial problems

Theorem: if Alg is a "greedy" algorithm, then the approximation ratio of Gittins+Alg+LH is ≤ 4/3 times that of Alg

Our contribution



Local Hedging (LH)

New *decomposition-based* technique for optional inspection

- Reduces problem to required-inspection case
- Naturally generalizes to combinatorial problems

Theorem: if Alg is a "greedy" algorithm, then the approximation ratio of **Gittins**+Alg+**LH** is ≤ 4/3 times that of Alg

price of reduction
from 3/4 discount

Our contribution

Key idea: randomization for "context-robustness"

Local Hedging (LH)

New *decomposition-based* technique for optional inspection

- Reduces problem to required-inspection case
- Naturally generalizes to combinatorial problems

NEW!

Theorem: if Alg is a "greedy" algorithm, then the approximation ratio of Gittins+Alg+LH is ≤ 4/3 times that of Alg

price of reduction from 3/4 discount

What is R, and how does it summarize $n - 1$ boxes?

What is R, and how does it summarize $n - 1$ boxes?

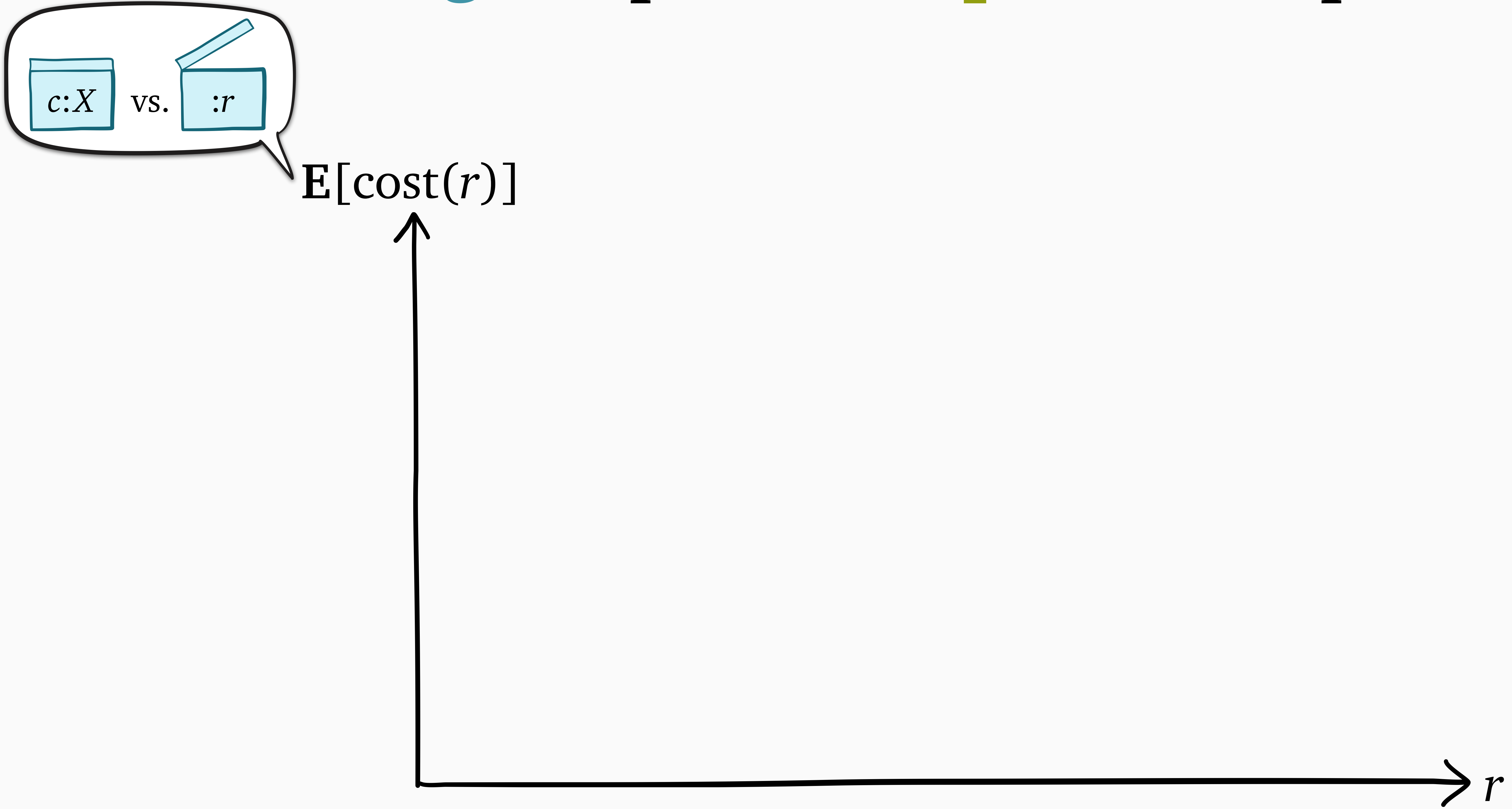
How does 3/4 discount in **Local Hedging** lemma become 4/3 approximation ratio?

What is R, and how does it summarize $n - 1$ boxes?

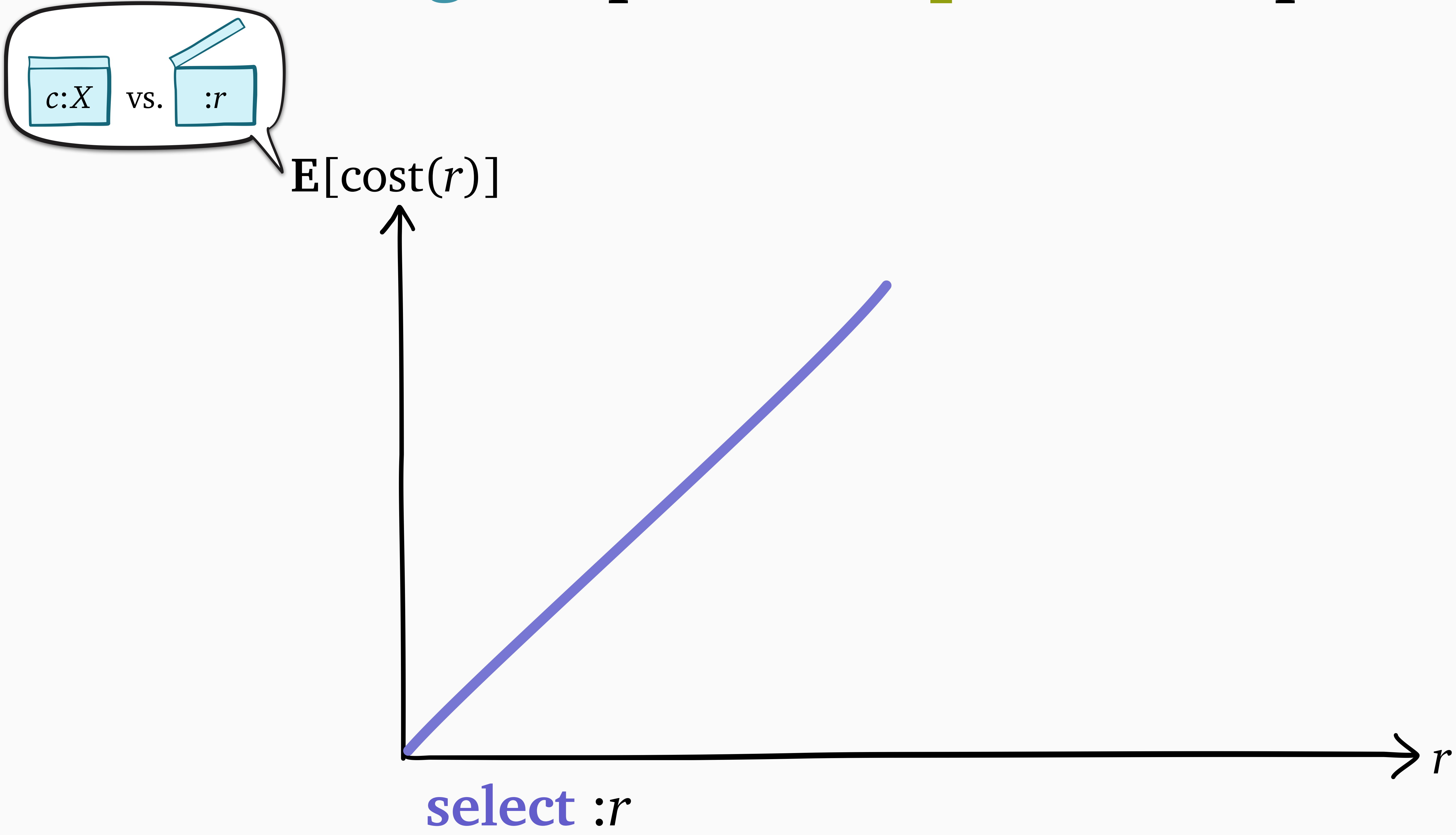
How does 3/4 discount in **Local Hedging** lemma become 4/3 approximation ratio?

Technical tool:
surrogate prices

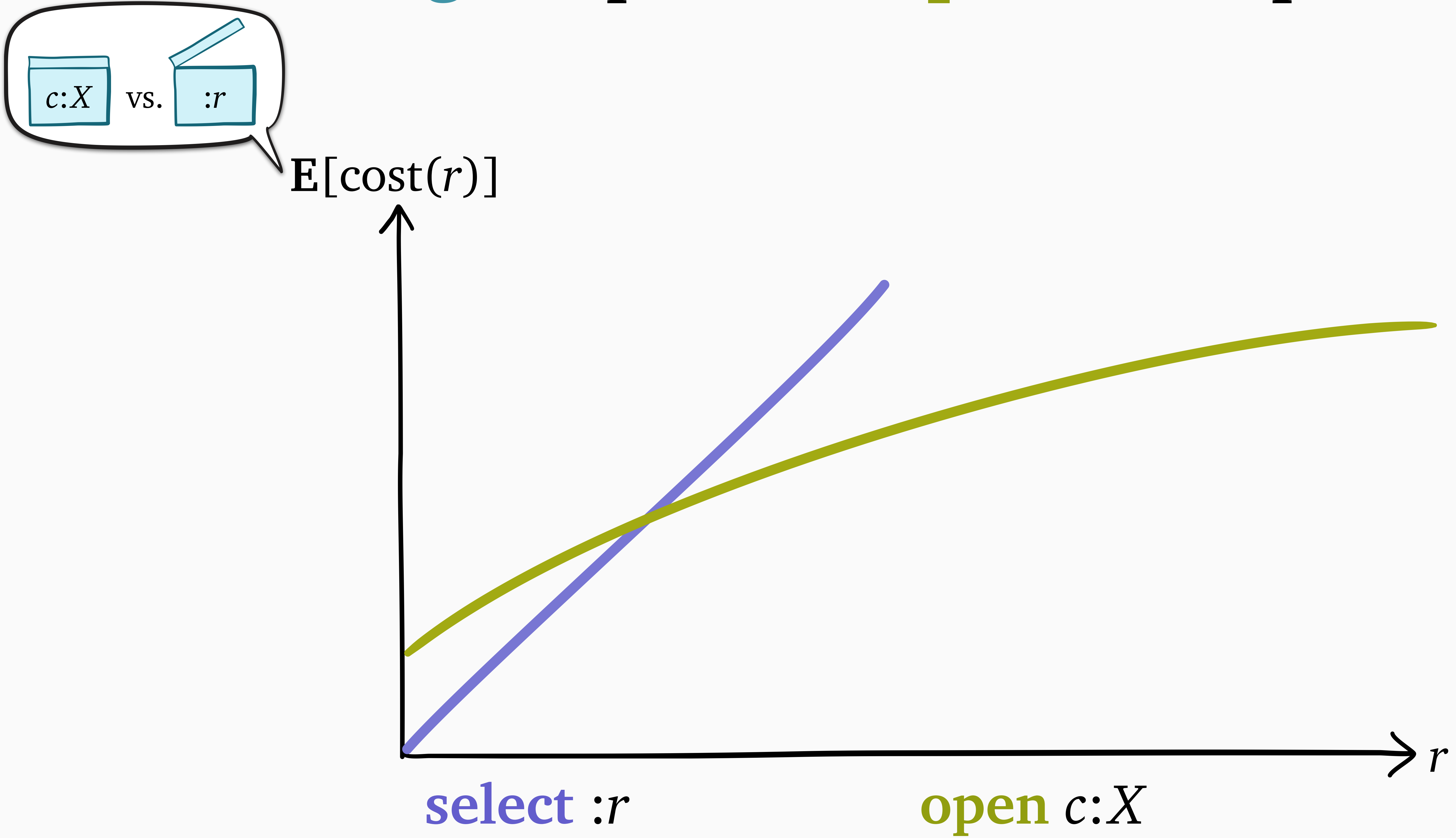
Surrogate price: required inspection



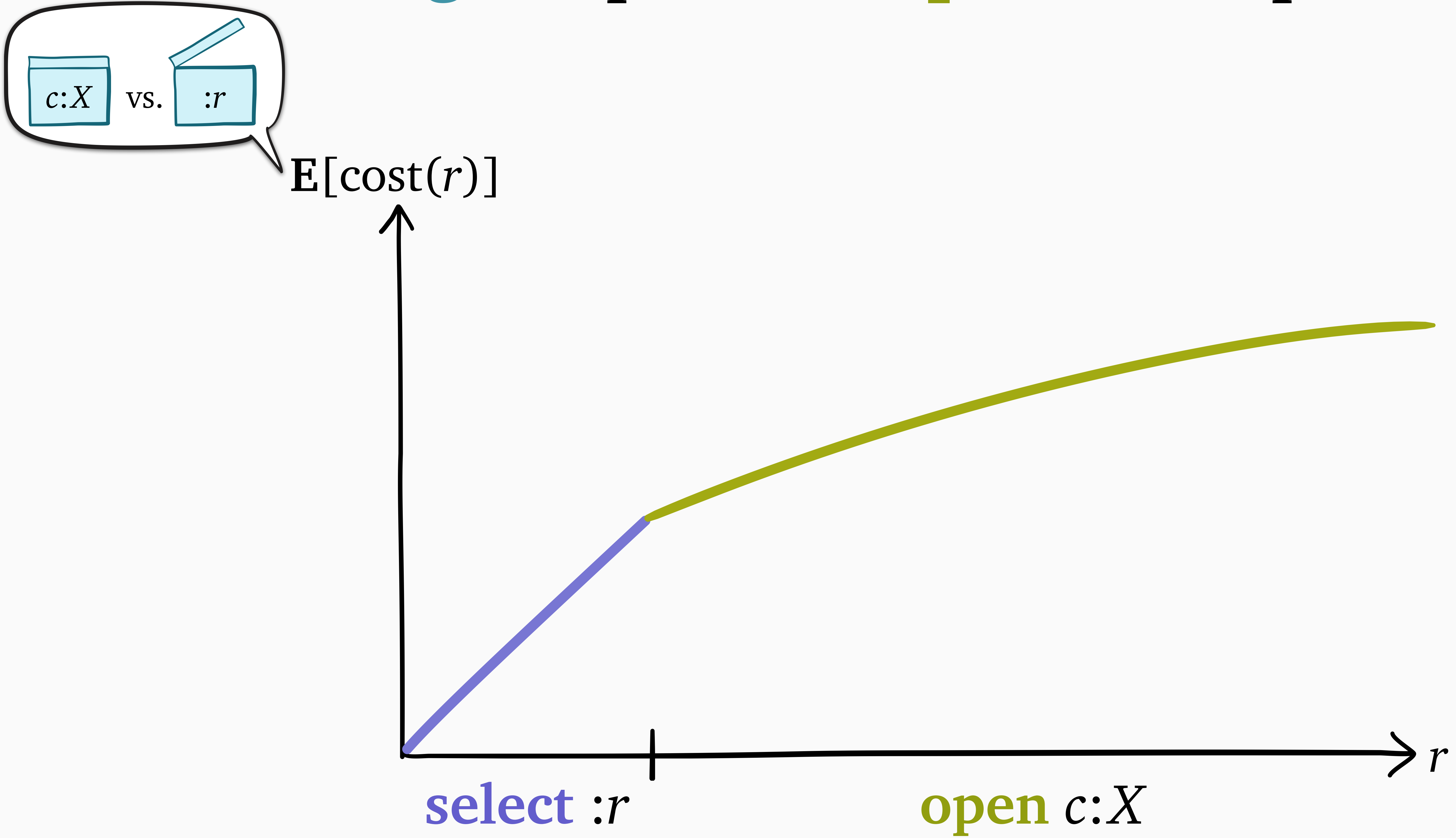
Surrogate price: required inspection



Surrogate price: required inspection



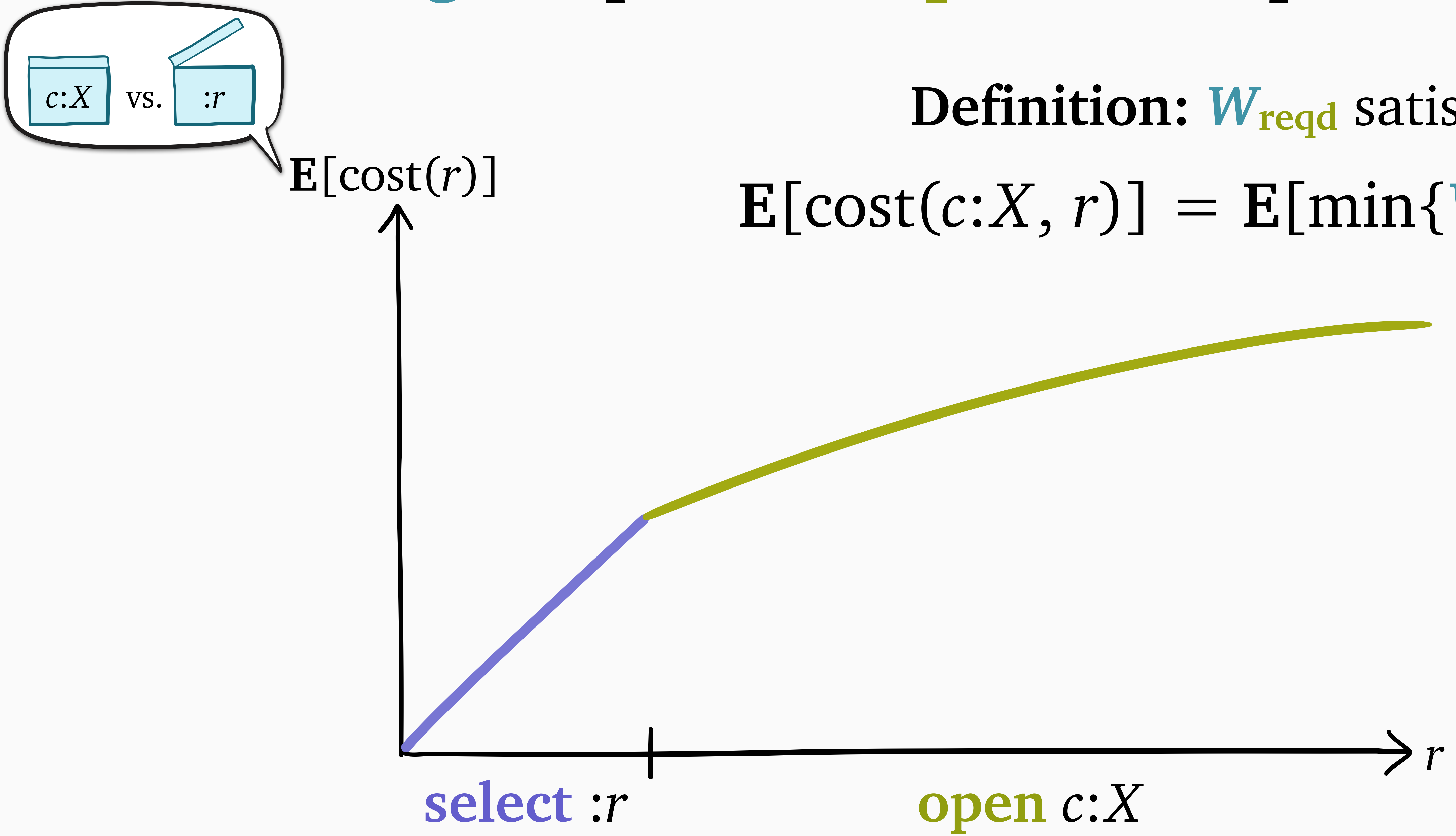
Surrogate price: required inspection



Surrogate price: required inspection

Definition: W_{reqd} satisfies

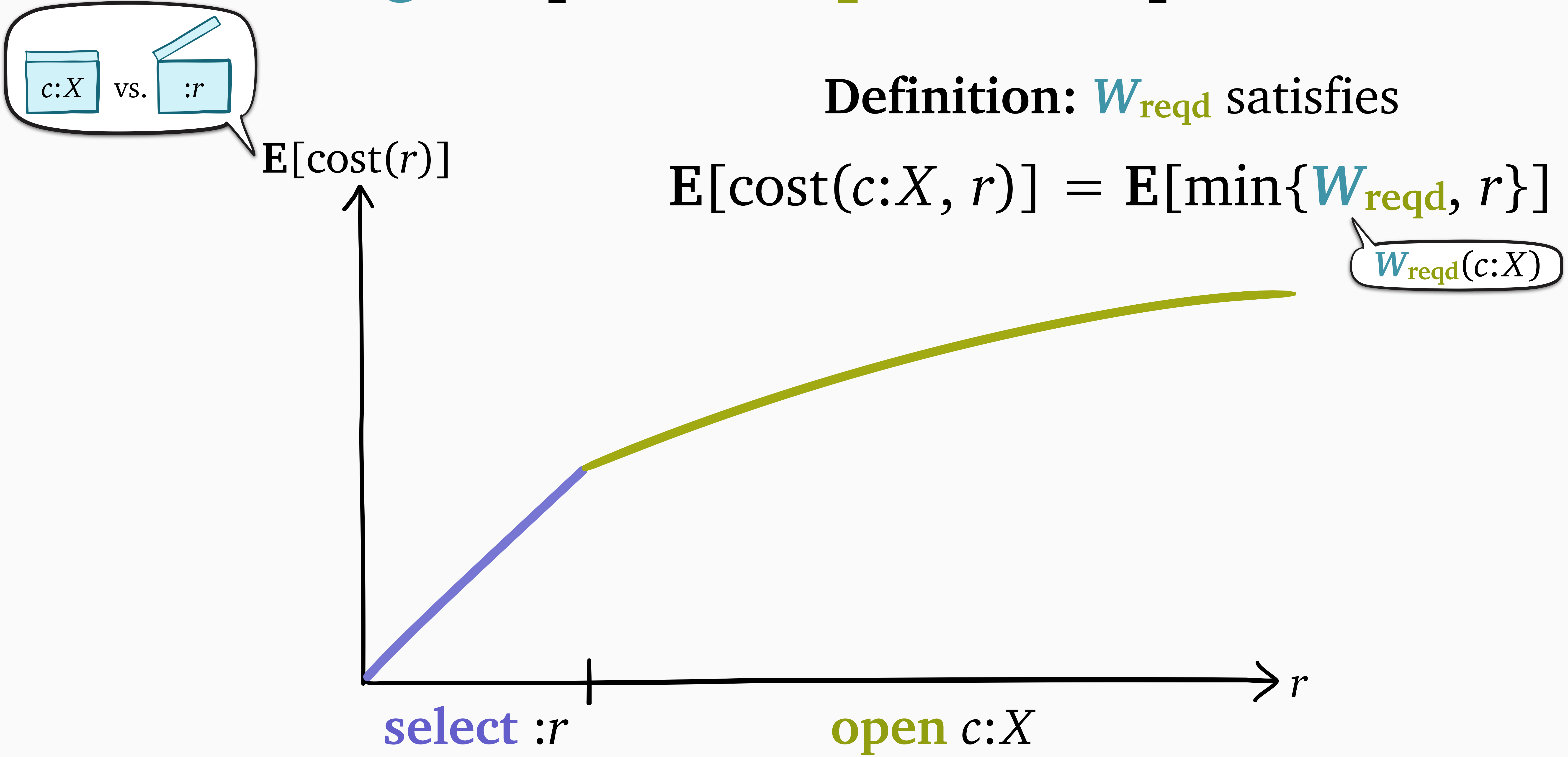
$$\mathbf{E}[\text{cost}(c{:}X, r)] = \mathbf{E}[\min\{W_{\text{reqd}}, r\}]$$



Surrogate price: required inspection

Definition: W_{reqd} satisfies

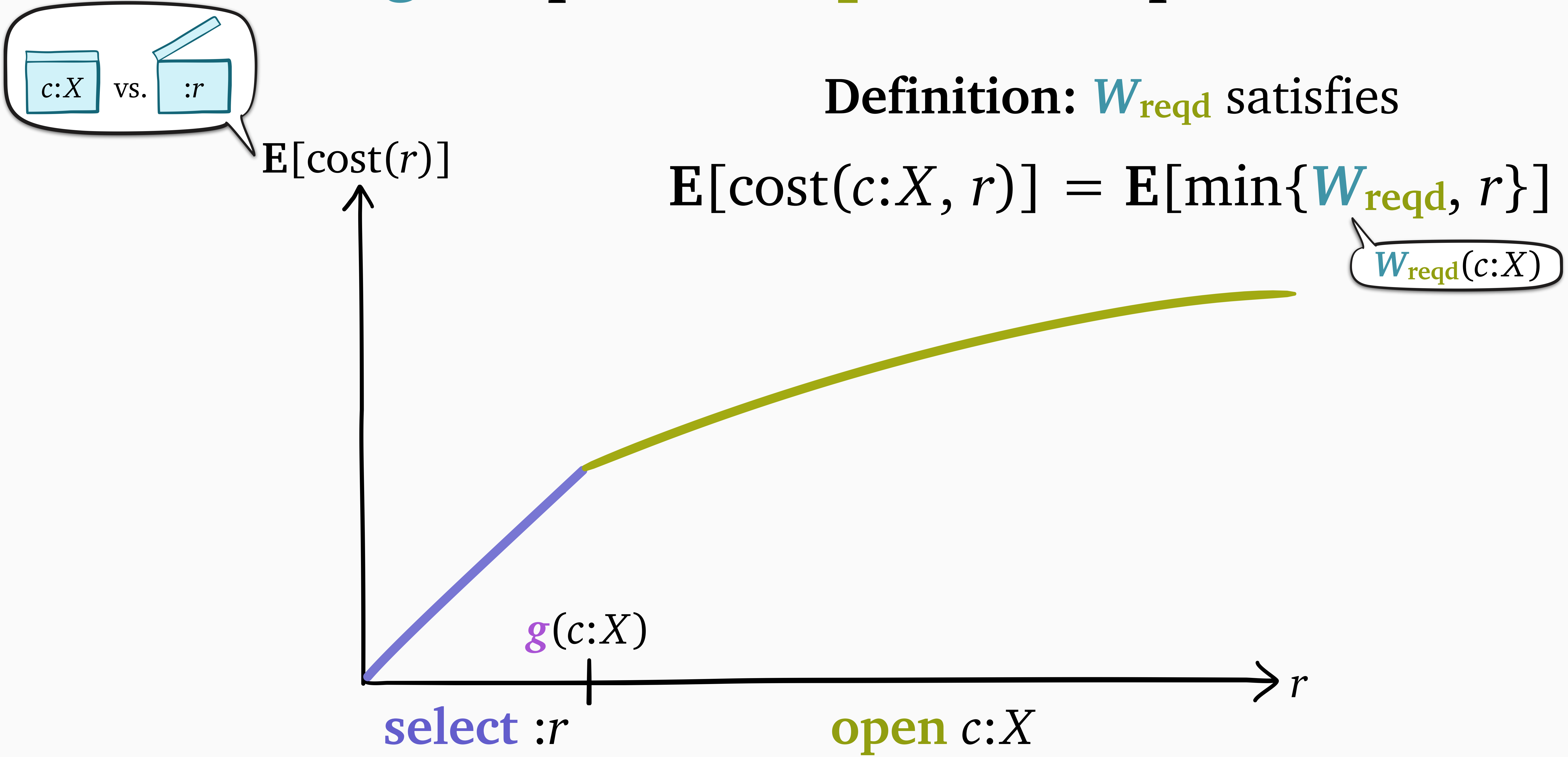
$$\mathbf{E}[\mathrm{cost}(c{:}X, r)] = \mathbf{E}[\min\{W_{\mathrm{reqd}}, r\}]$$



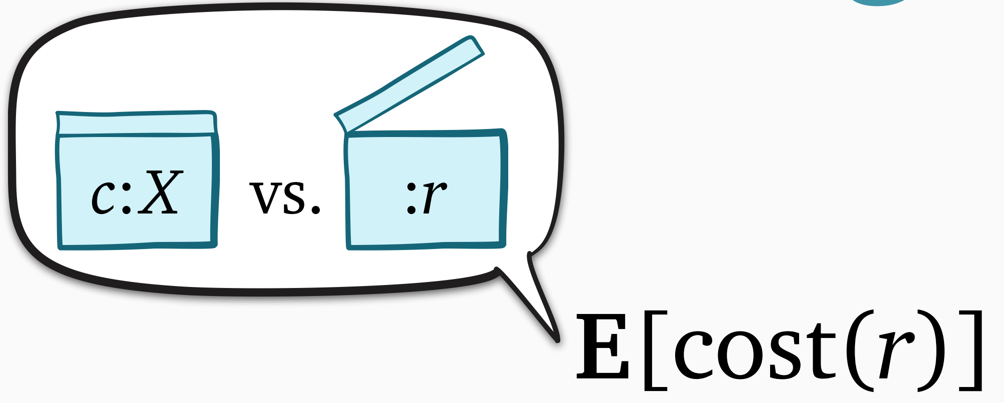
Surrogate price: required inspection

Definition: W_{reqd} satisfies

$$\mathbf{E}[\text{cost}(c{:}X, r)] = \mathbf{E}[\min\{W_{\text{reqd}}, r\}]$$



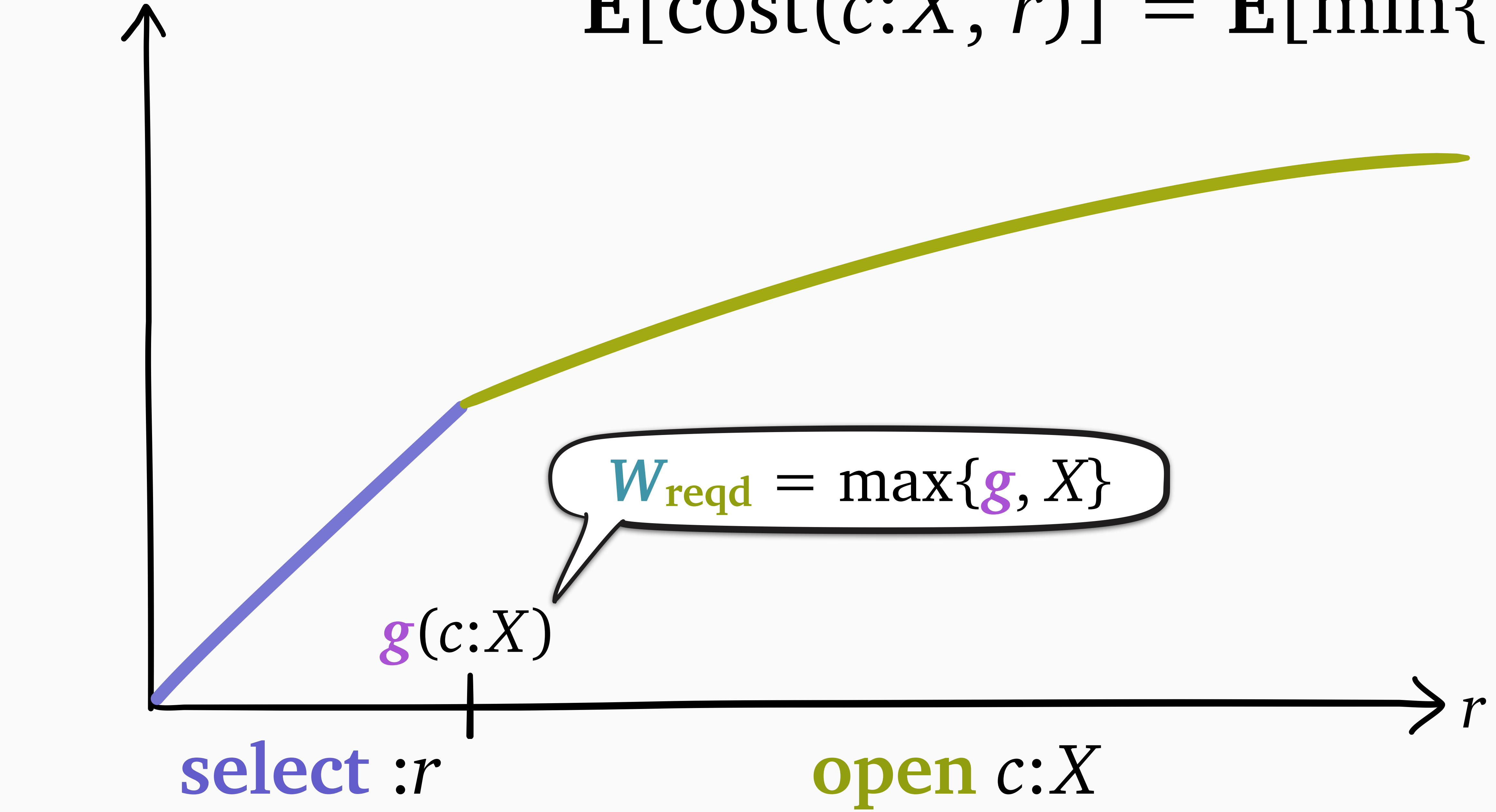
Surrogate price: required inspection



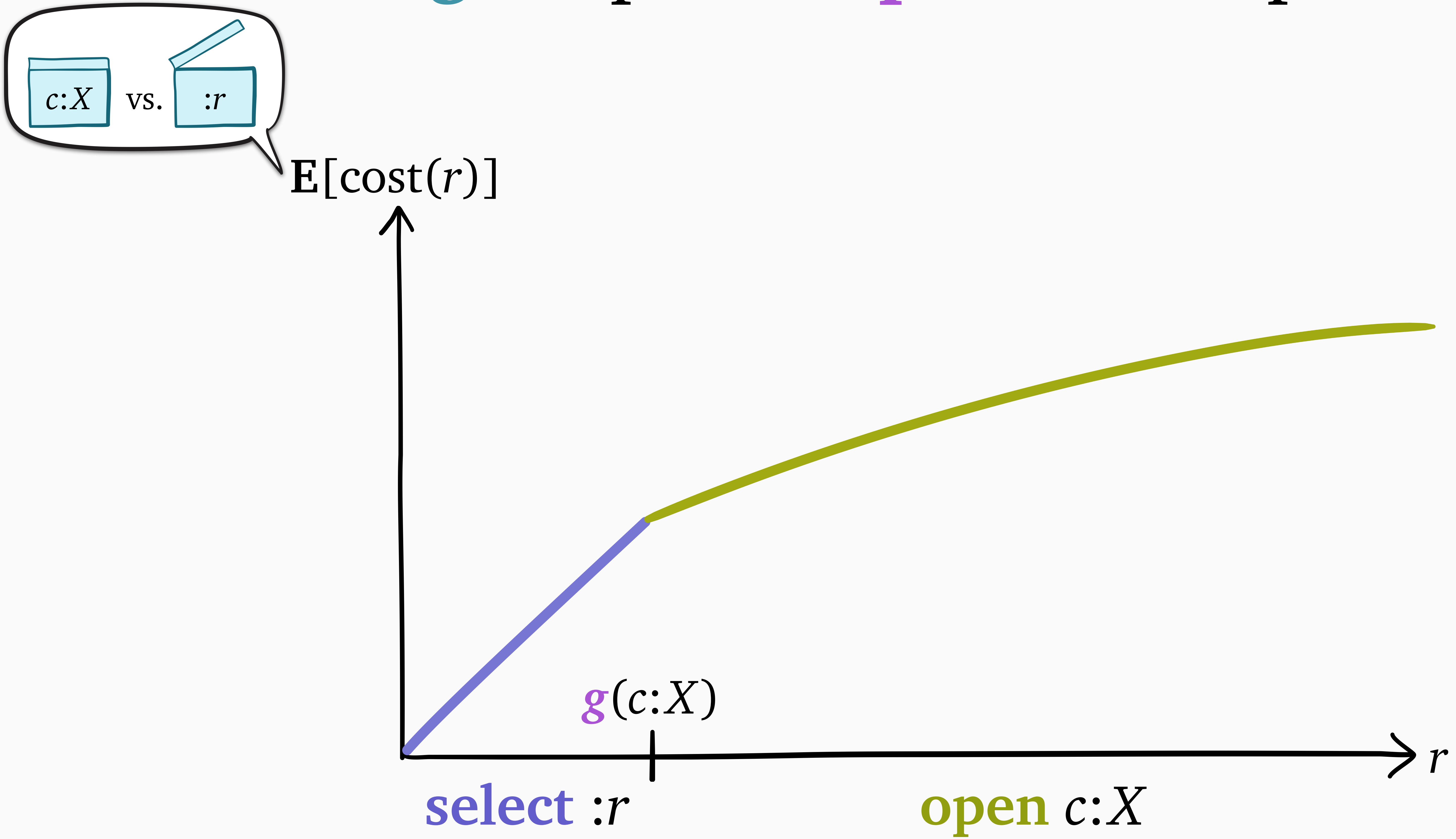
Definition: W_{reqd} satisfies

$$\mathbf{E}[\text{cost}(c{:}X,\ r)] = \mathbf{E}[\min\{W_{\text{reqd}},\ r\}]$$

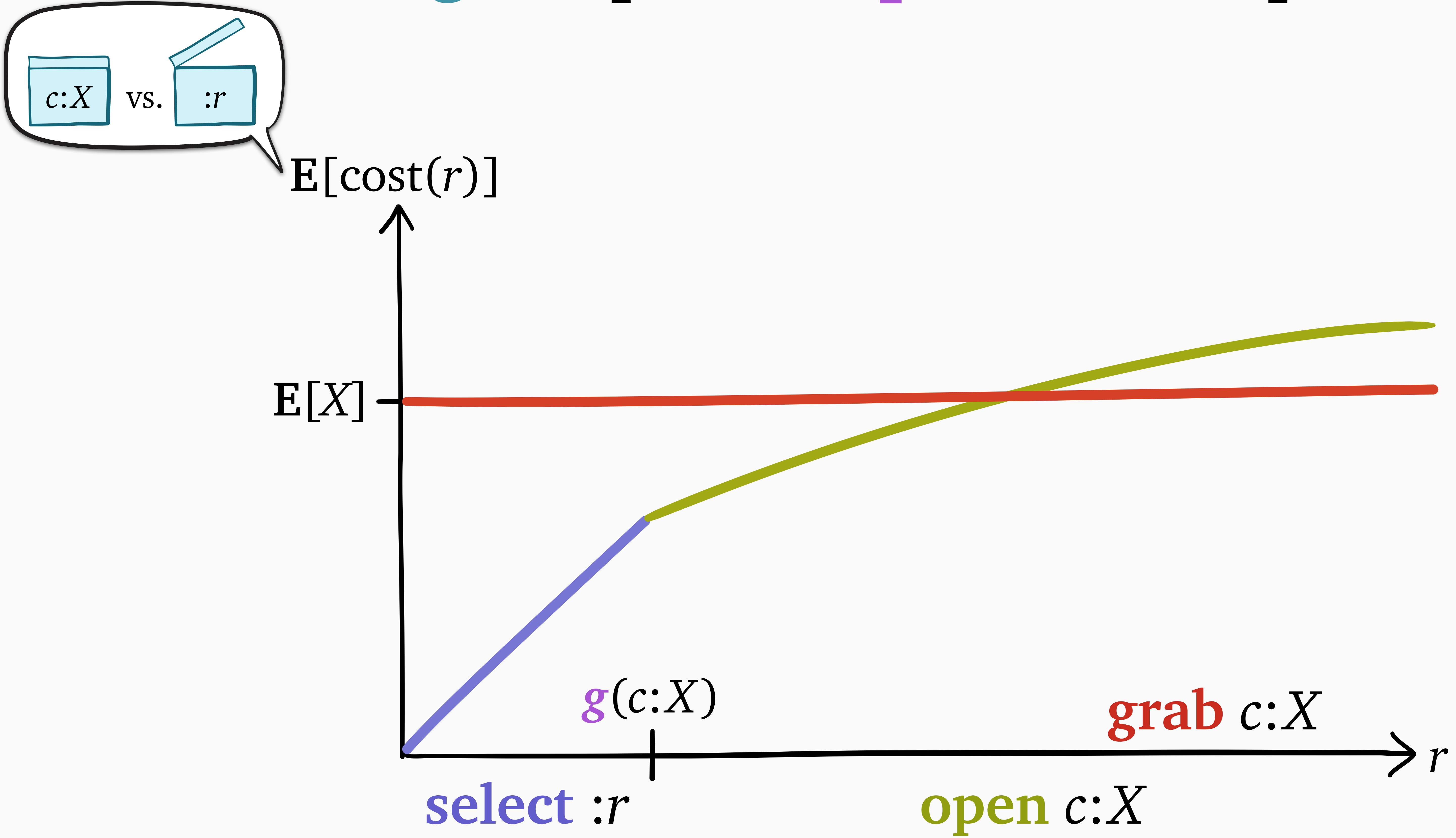
$W_{\text{reqd}}(c{:}X)$



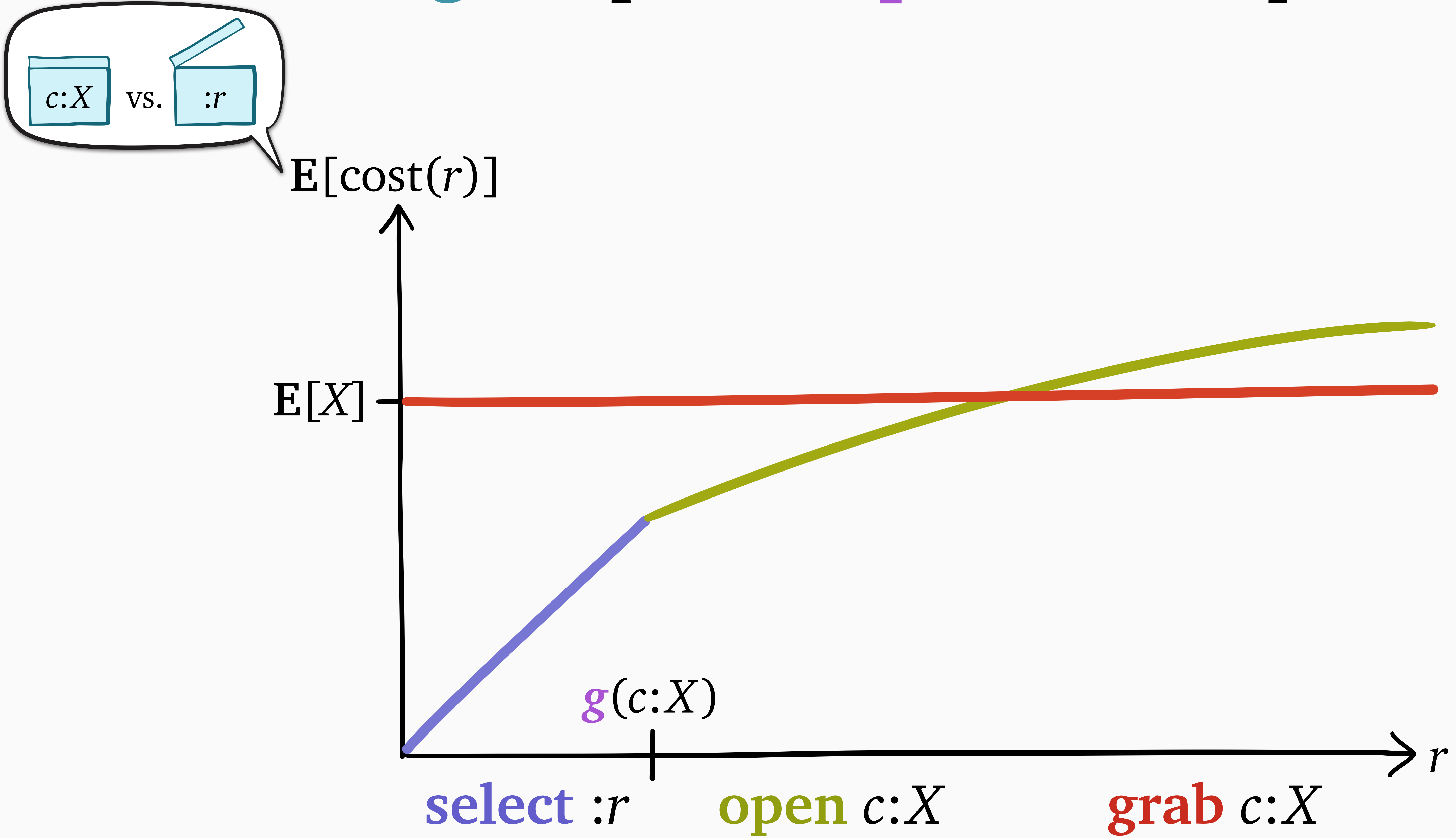
Surrogate price: optional inspection



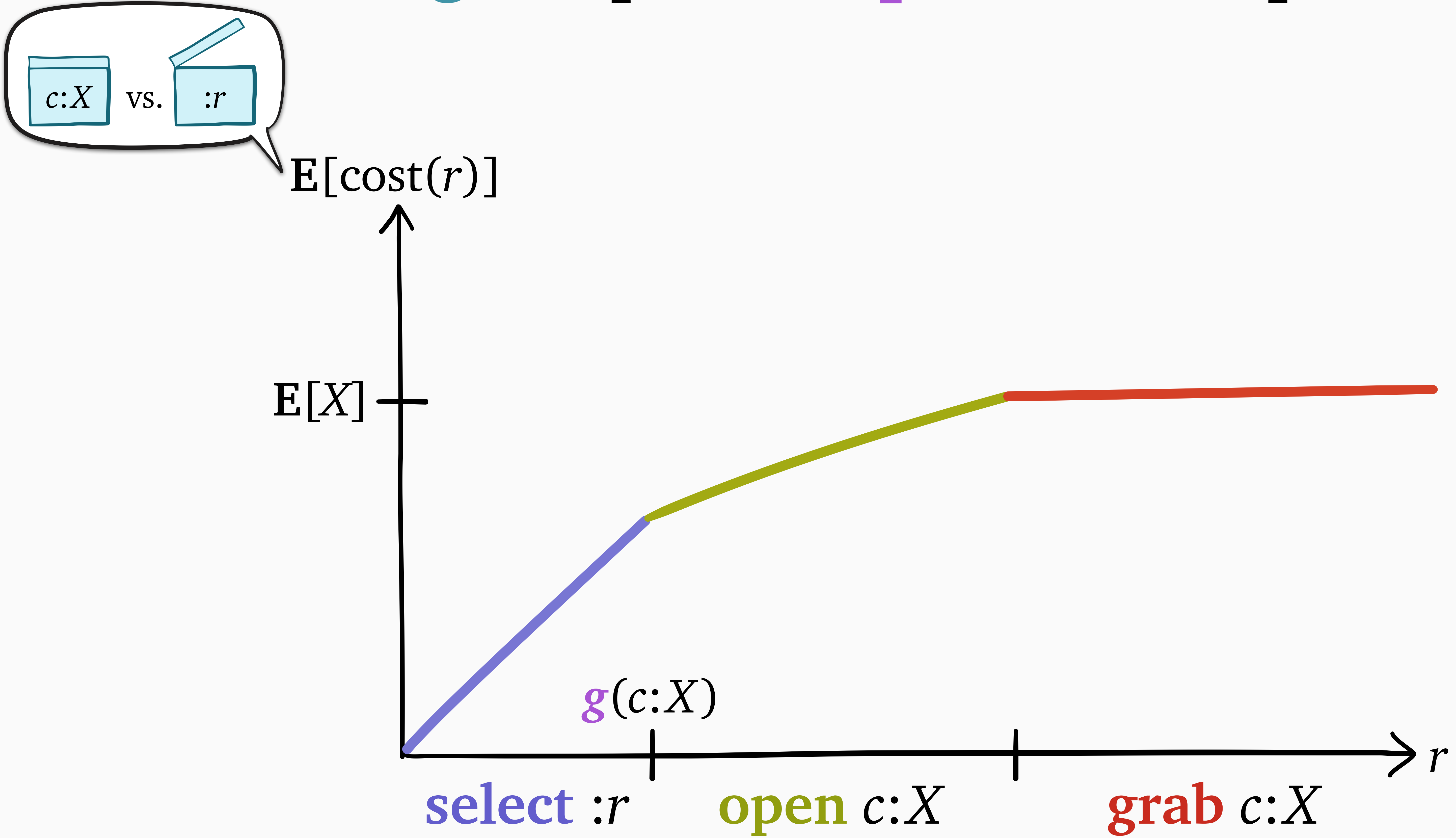
Surrogate price: optional inspection



Surrogate price: optional inspection



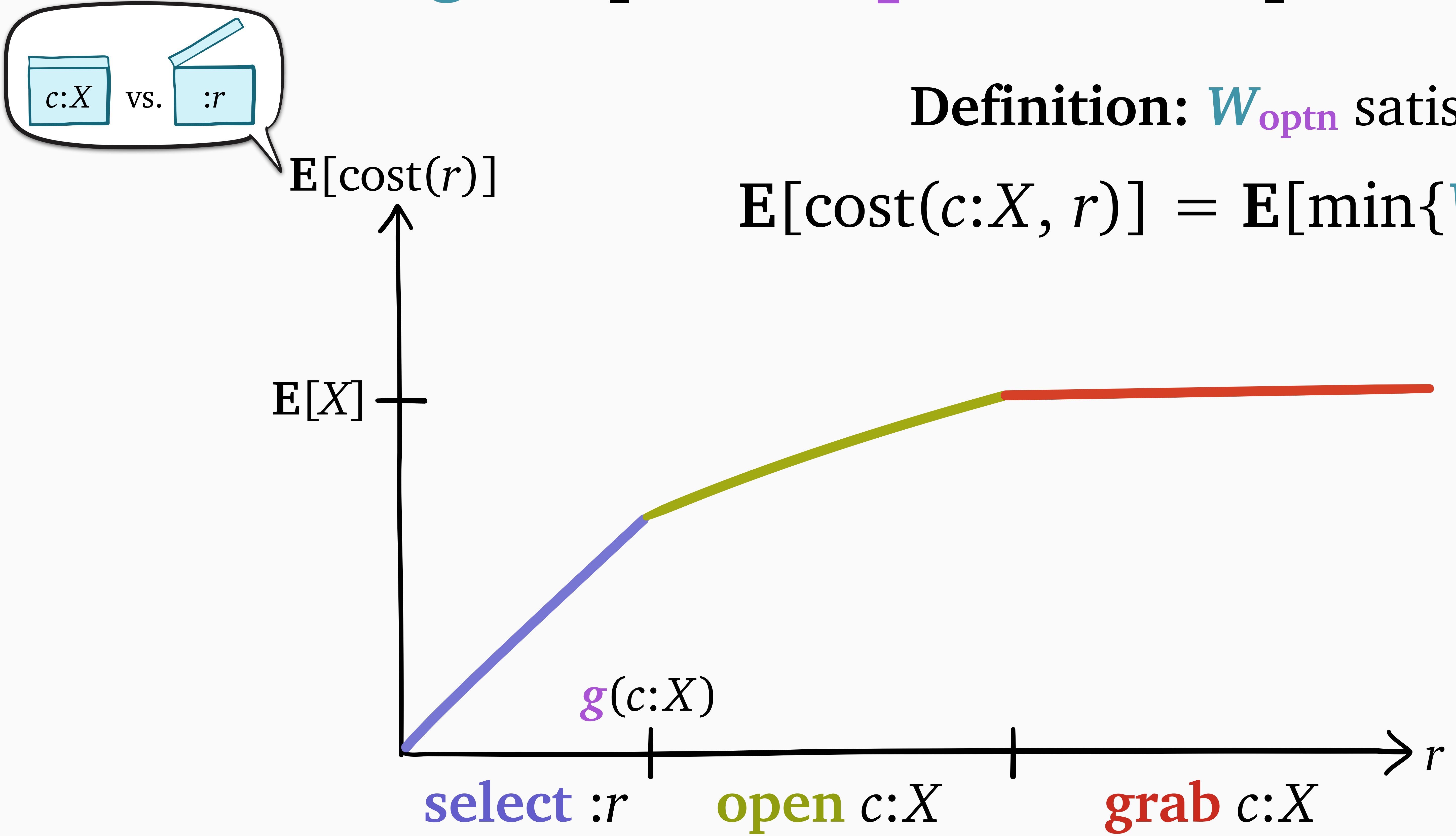
Surrogate price: optional inspection



Surrogate price: optional inspection

Definition: W_{optn} satisfies

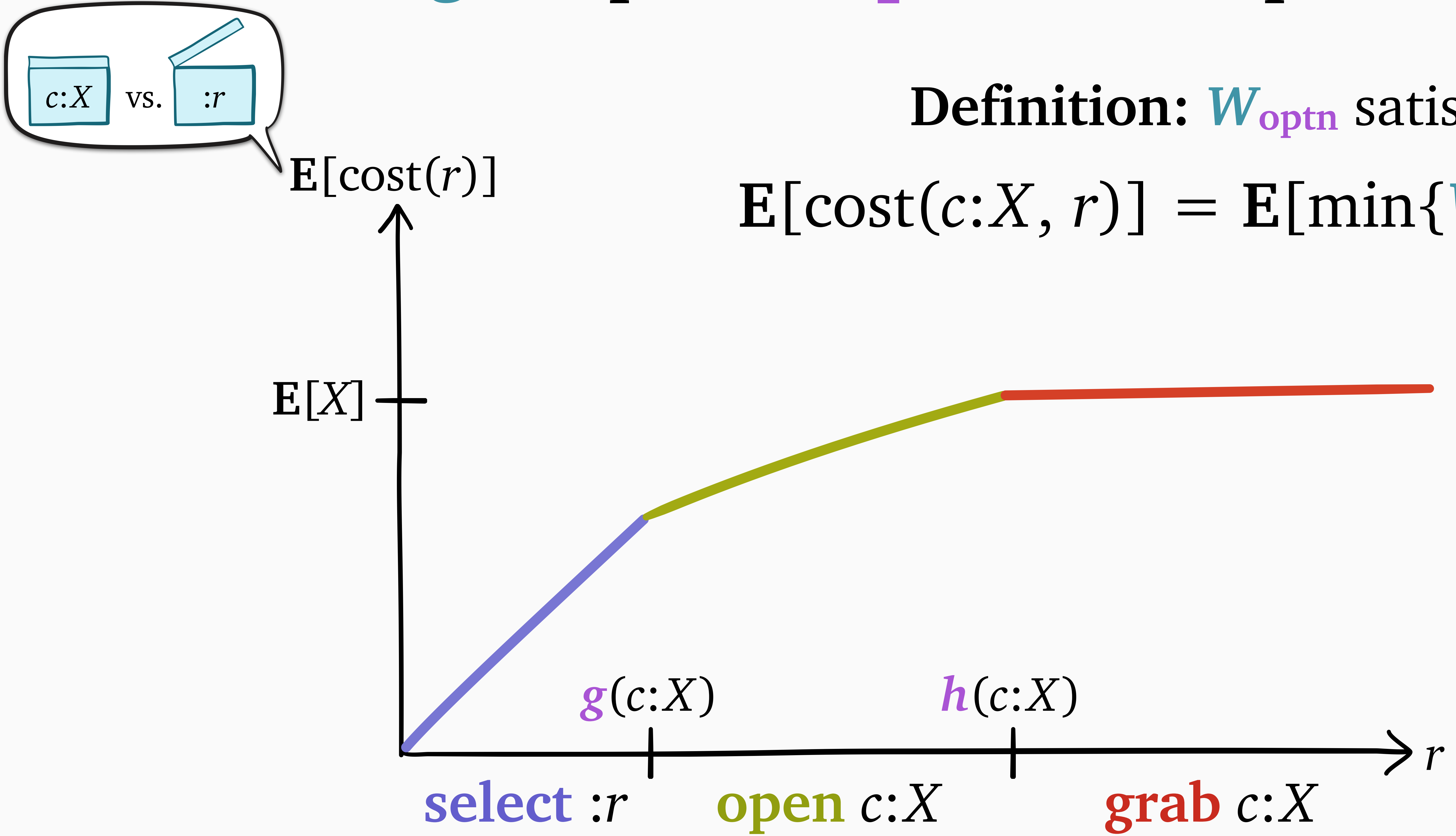
$$\mathbf{E}[\text{cost}(c{:}X, r)] = \mathbf{E}[\min\{W_{\text{optn}}, r\}]$$



Surrogate price: optional inspection

Definition: W_{optn} satisfies

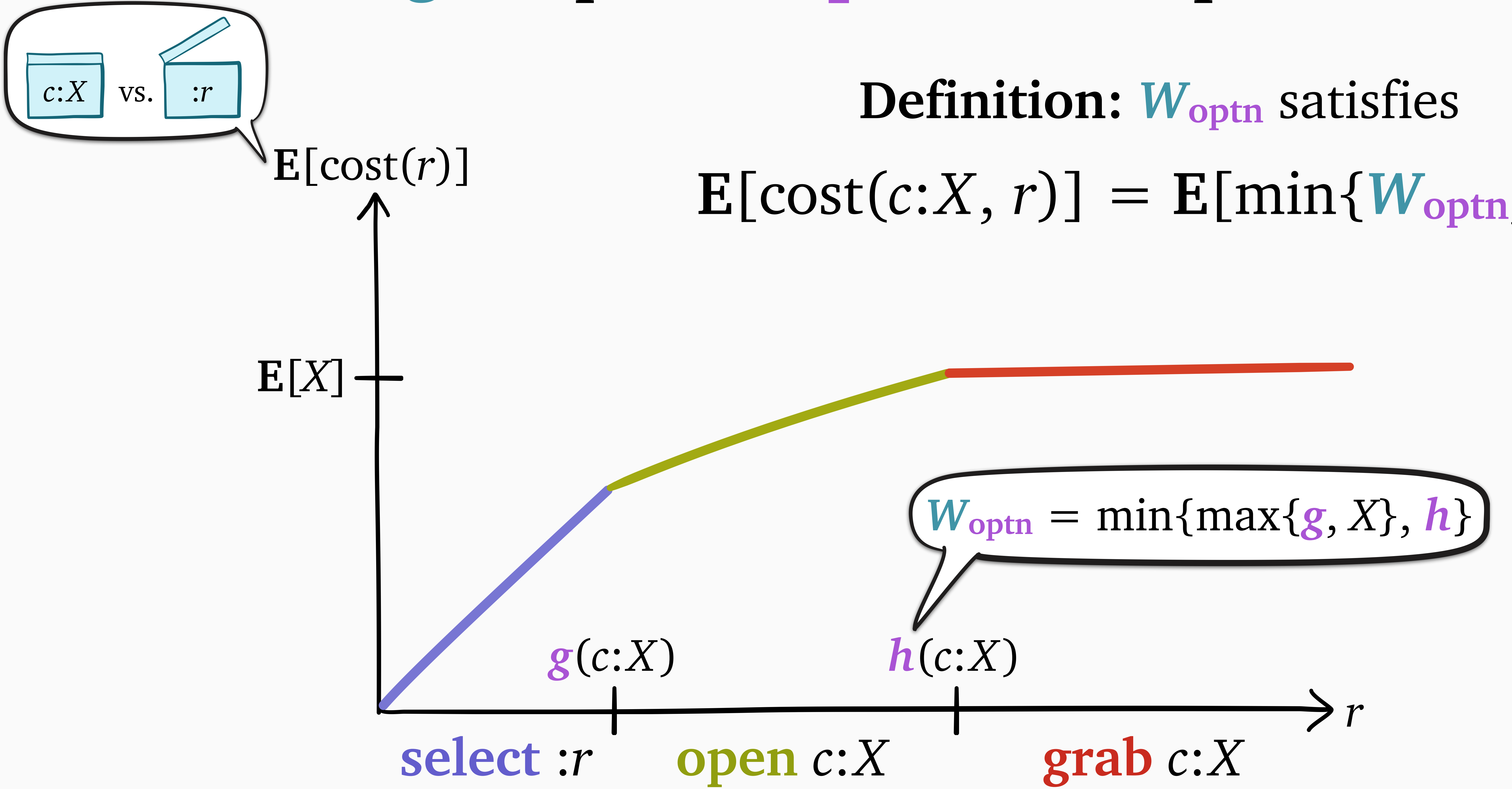
$$\mathbf{E}[\text{cost}(c{:}X, r)] = \mathbf{E}[\min\{W_{\text{optn}}, r\}]$$



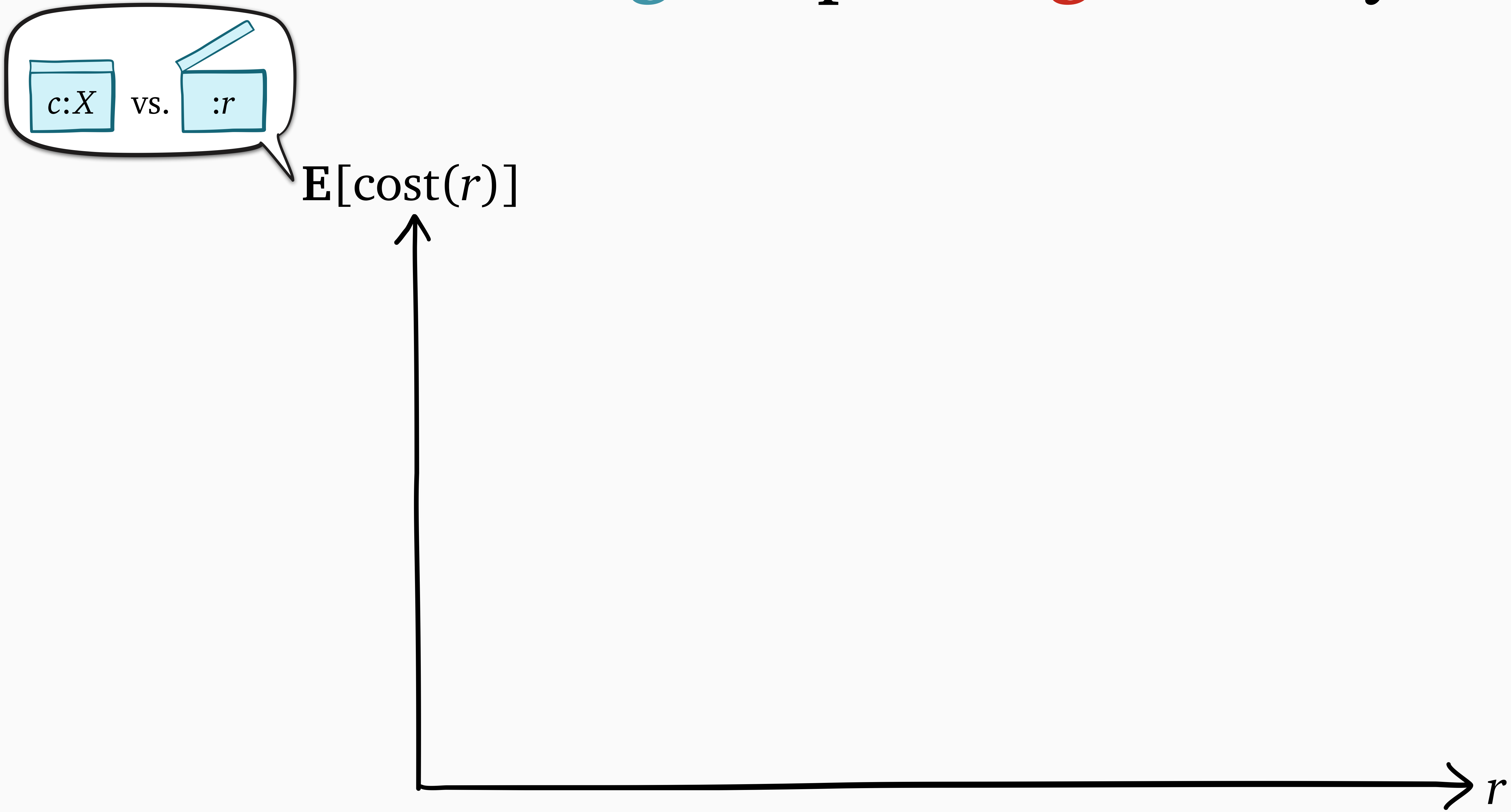
Surrogate price: optional inspection

Definition: W_{optn} satisfies

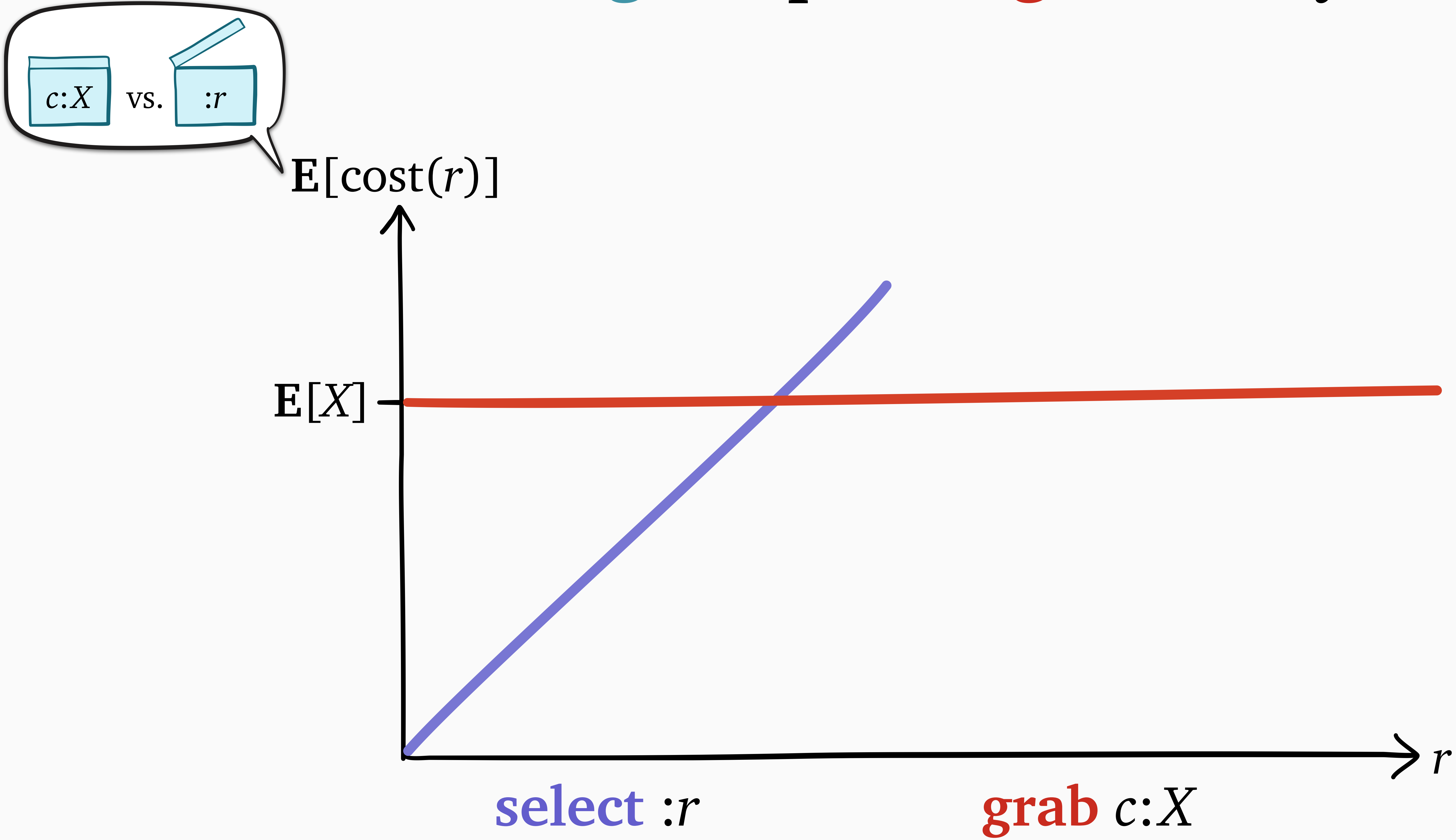
$$\mathbf{E}[\text{cost}(c{:}X, r)] = \mathbf{E}[\min\{W_{\text{optn}}, r\}]$$



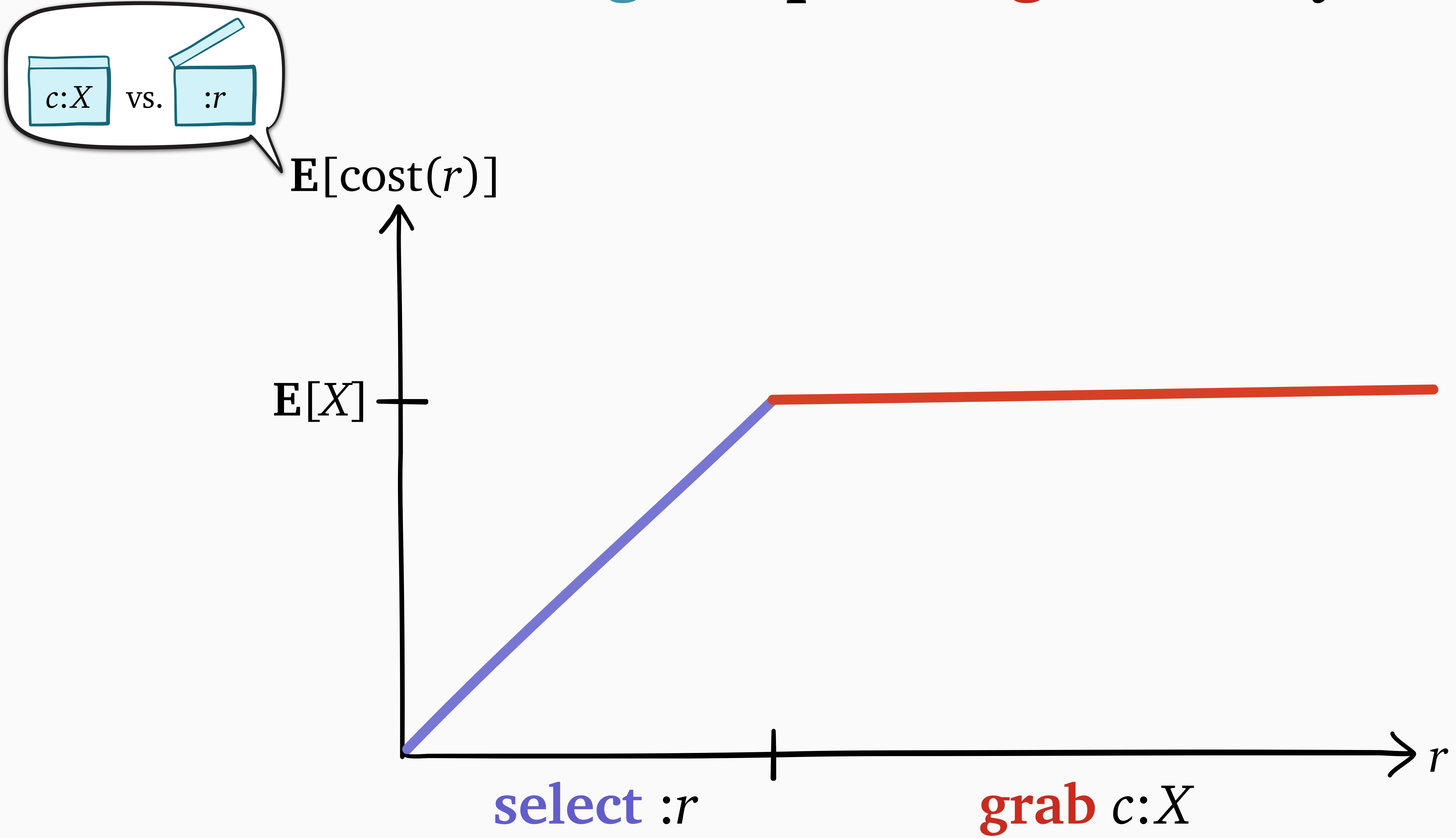
Surrogate price: grab only



Surrogate price: grab only



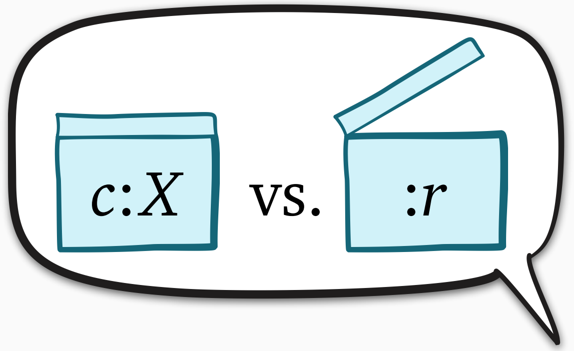
Surrogate price: grab only

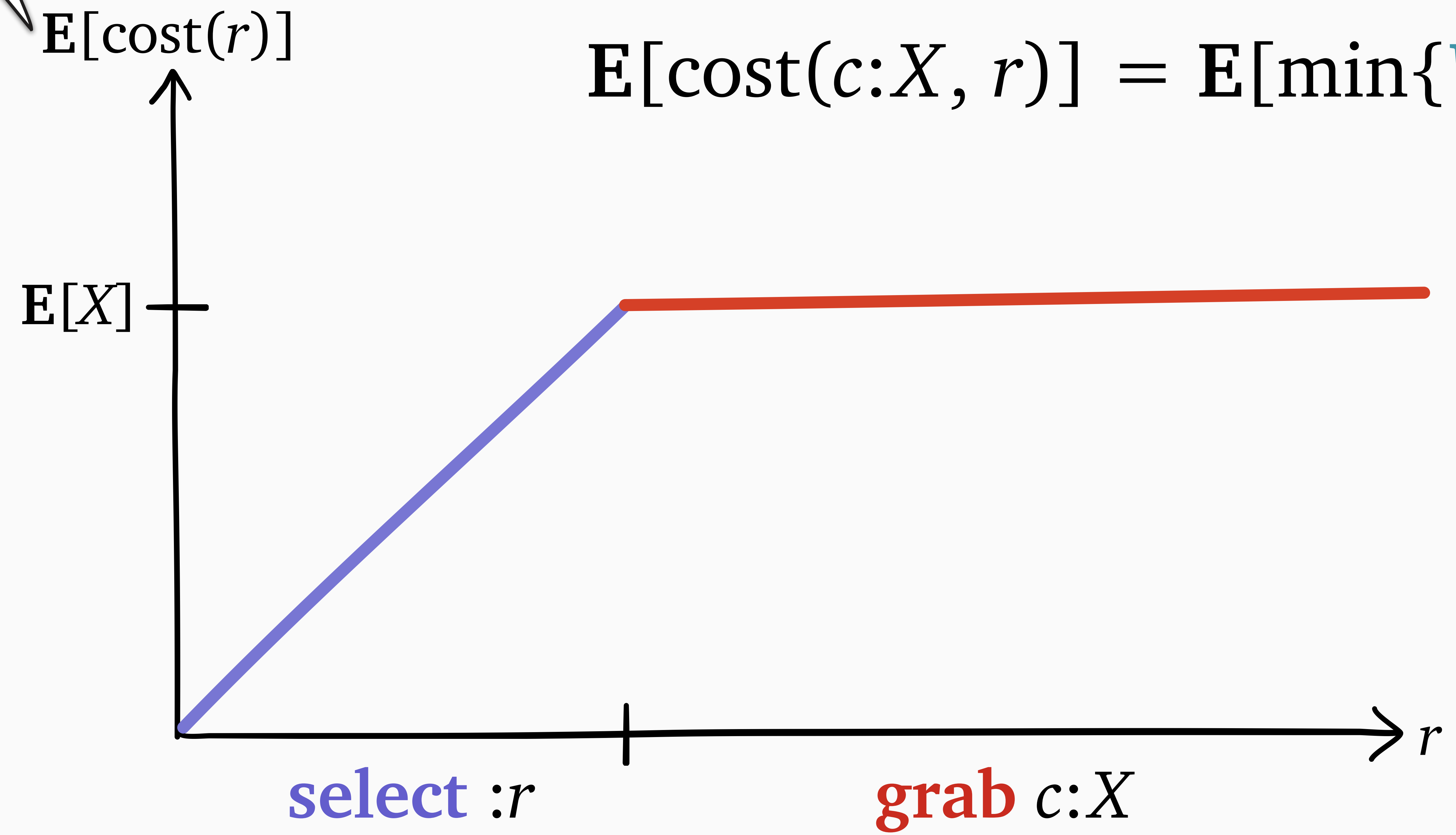


Surrogate price: grab only

Definition: W_{grab} satisfies

$$\mathbf{E}[\text{cost}(c{:}X, r)] = \mathbf{E}[\min\{W_{\text{grab}}, r\}]$$

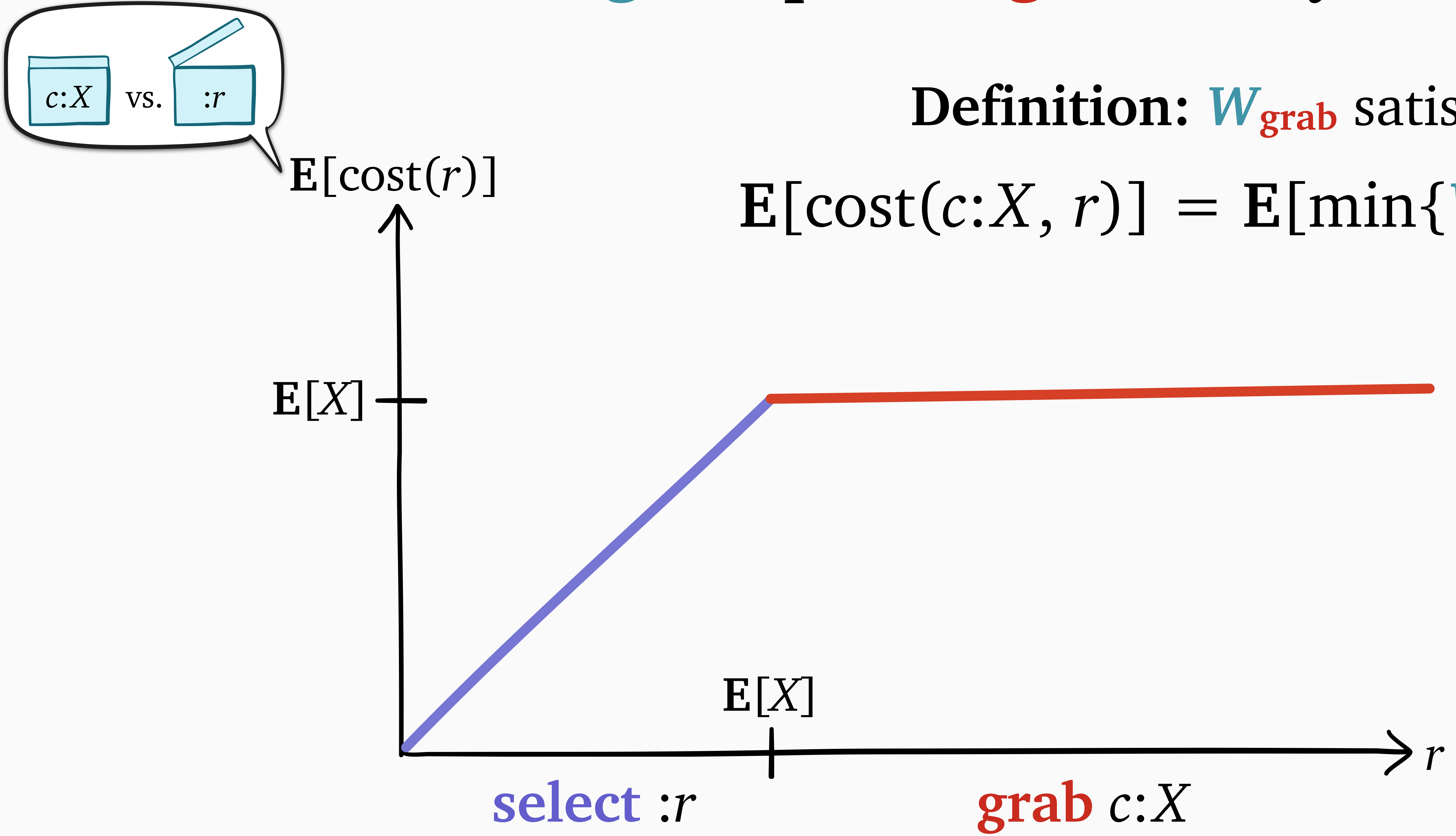




Surrogate price: grab only

Definition: W_{grab} satisfies

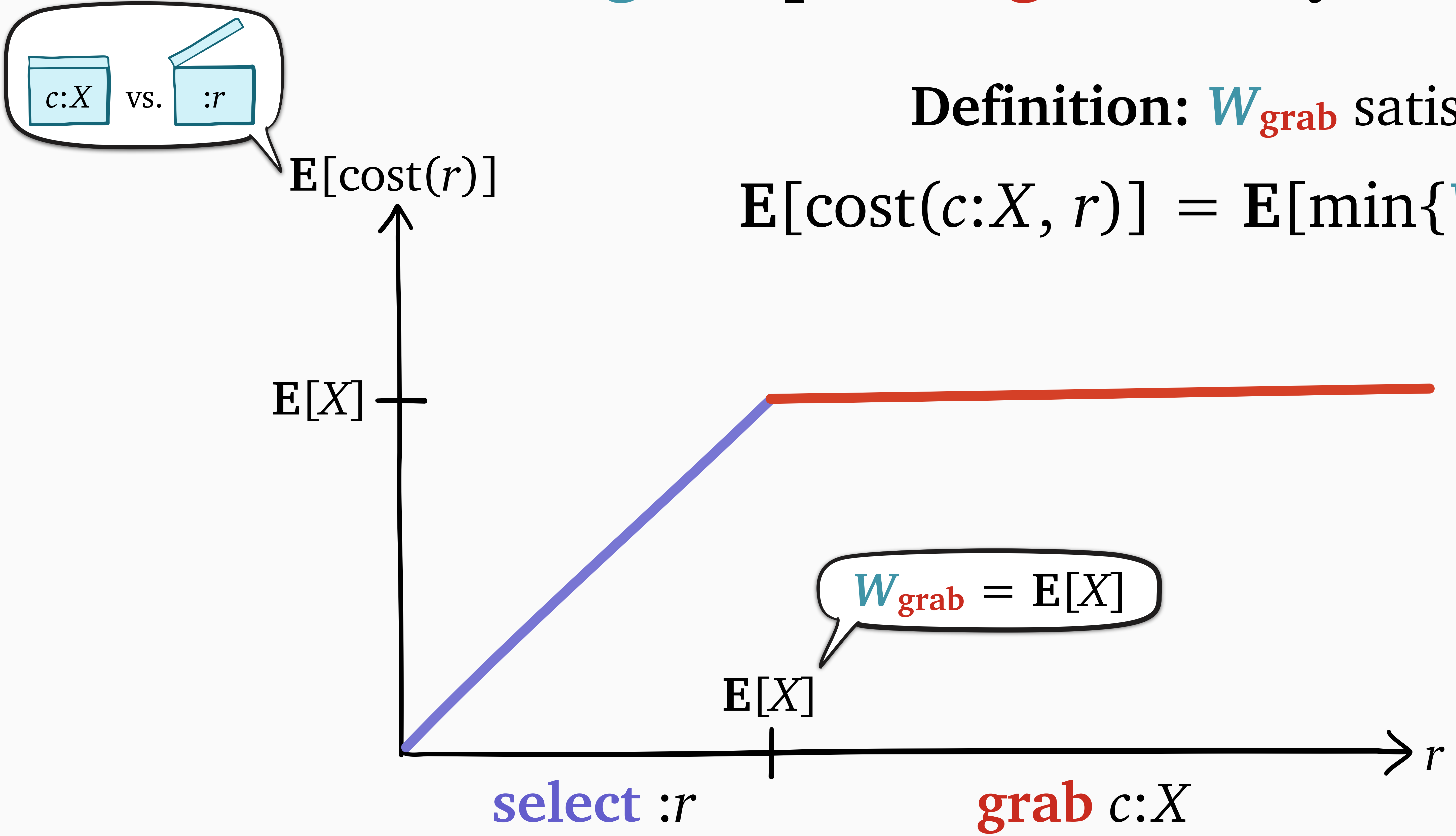
$$\mathbf{E}[\text{cost}(c{:}X, r)] = \mathbf{E}[\min\{W_{\text{grab}}, r\}]$$



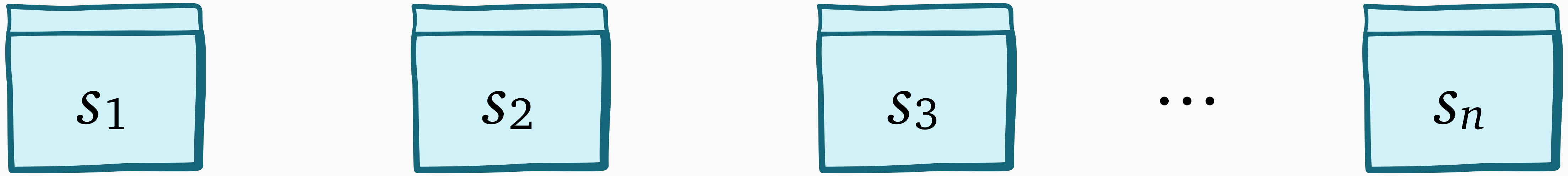
Surrogate price: grab only

Definition: W_{grab} satisfies

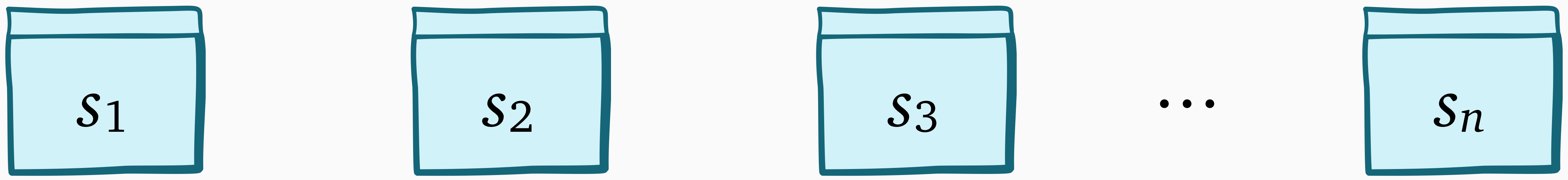
$$\mathbf{E}[\text{cost}(c{:}X,\, r)] = \mathbf{E}[\min\{W_{\text{grab}},\, r\}]$$



Expressing cost using **surrogate** prices

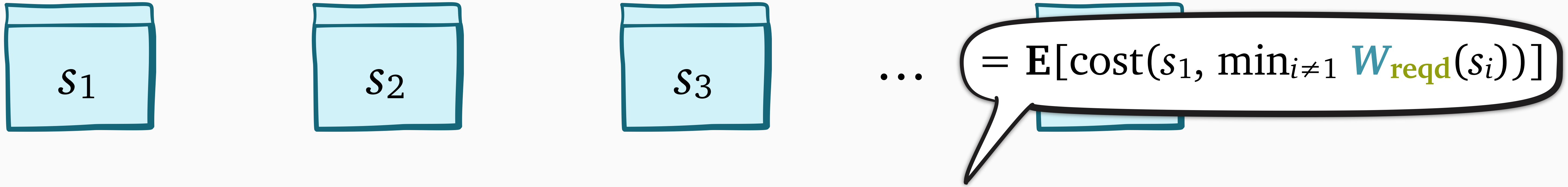


Expressing cost using **surrogate** prices



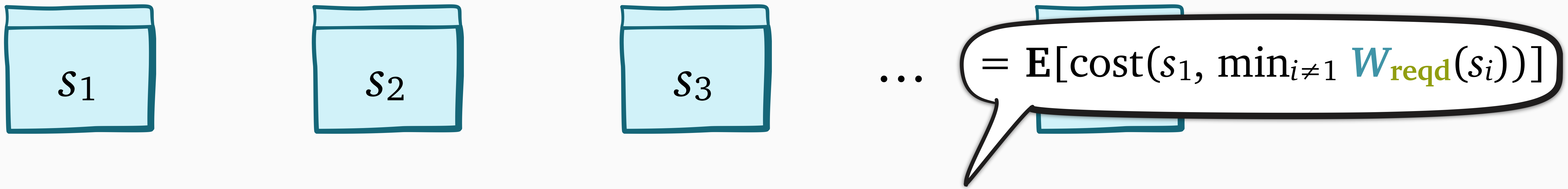
$$\mathbf{E}[\text{optimal } \textbf{required}\text{-inspection cost}] = \mathbf{E}[\min_i W_{\textbf{reqd}}(s_i)]$$

Expressing cost using surrogate prices



$$\mathbf{E}[\text{optimal required-inspection cost}] = \mathbf{E}[\min_i W_{\text{reqd}}(s_i)]$$

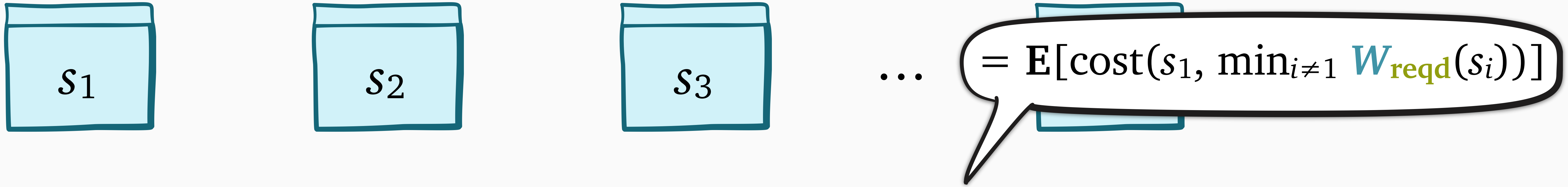
Expressing cost using **surrogate** prices



$$\mathbf{E}[\text{optimal } \textbf{required}\text{-inspection cost}] = \mathbf{E}[\min_i W_{\text{reqd}}(s_i)]$$

$$\mathbf{E}[\text{optimal } \textbf{optional}\text{-inspection cost}] \geq \mathbf{E}[\min_i W_{\text{optn}}(s_i)]$$

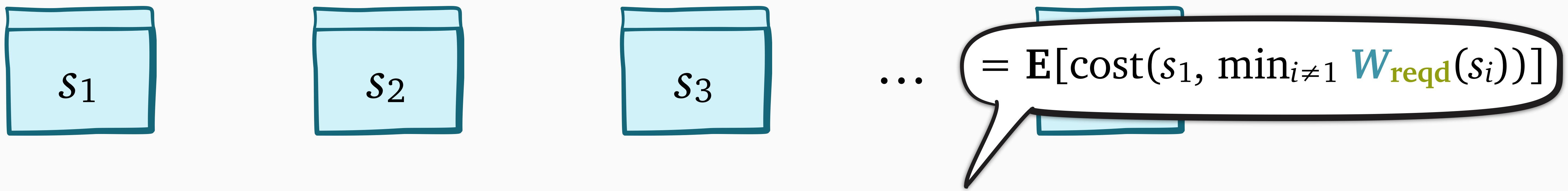
Expressing cost using surrogate prices



$$\mathbf{E}[\text{optimal } \textbf{required}\text{-inspection cost}] = \mathbf{E}[\min_i W_{\textbf{reqd}}(s_i)]$$

newish $$\mathbf{E}[\text{optimal } \textbf{optional}\text{-inspection cost}] \geq \mathbf{E}[\min_i W_{\textbf{optn}}(s_i)]$$

Expressing cost using surrogate prices

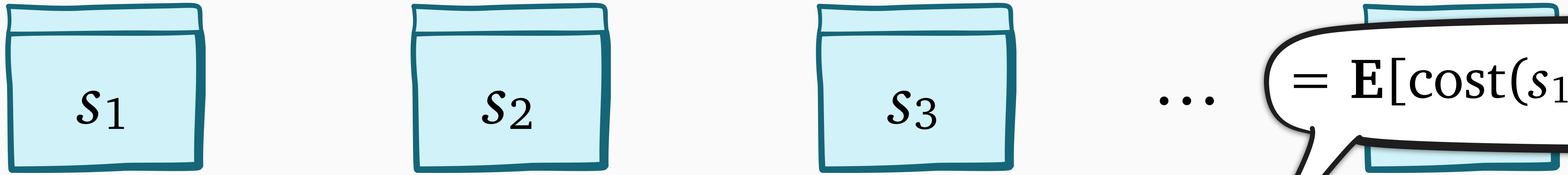


$\mathbf{E}[\text{optimal } \textbf{required}\text{-inspection cost}] = \mathbf{E}[\min_i W_{\mathrm{reqd}}(s_i)]$

newish $\mathbf{E}[\text{optimal } \textbf{optional}\text{-inspection cost}] \geq \mathbf{E}[\min_i W_{\mathrm{optn}}(s_i)]$

$\mathbf{E}[\text{cost under } \textbf{Local Hedging}] = \mathbf{E}[\min_i W_{\mathrm{LH}}(s_i)]$

Expressing cost using surrogate prices



$= \mathbf{E}[\text{cost}(s_1, \min_{i \neq 1} W_{\text{reqd}}(s_i))]$

$\mathbf{E}[\text{optimal required-inspection cost}] = \mathbf{E}[\min_i W_{\text{reqd}}(s_i)]$

newish $\mathbf{E}[\text{optimal optional-inspection cost}] \geq \mathbf{E}[\min_i W_{\text{optn}}(s_i)]$

$\mathbf{E}[\text{cost under Local Hedging}] = \mathbf{E}[\min_i W_{\text{LH}}(s_i)]$

$$W_{\text{LH}} = \begin{cases} W_{\text{reqd}} & \text{w.p. } p \\ W_{\text{grab}} & \text{w.p. } 1-p \end{cases}$$

Expressing cost using **surrogate** prices

s_1 s_2 s_3 …

$= \mathbf{E}[\text{cost}(s_1, \min_{i \neq 1} W_{\text{reqd}}(s_i))]$

$\mathbf{E}[\text{optimal } \textbf{required}\text{-inspection cost}] = \mathbf{E}[\min_i W_{\text{reqd}}(s_i)]$

newish $\mathbf{E}[\text{optimal } \textbf{optional}\text{-inspection cost}] \geq \mathbf{E}[\min_i W_{\text{optn}}(s_i)]$

$\mathbf{E}[\text{cost under } \textbf{Local Hedging}] = \mathbf{E}[\min_i W_{\text{LH}}(s_i)]$

NEW! **Lemma:** exists prob p s.t. for all r,

$$\mathbf{E}[\min\{\tfrac{3}{4} W_{\text{LH}}, r\}] \leq \mathbf{E}[\min\{W_{\text{optn}}, r\}]$$

$$W_{\text{LH}} = \begin{cases} W_{\textbf{reqd}} & \text{w.p. } p \\ W_{\textbf{grab}} & \text{w.p. } 1-p \end{cases}$$

Expressing cost using **surrogate** prices

s_1 s_2 s_3 …

$= \mathbf{E}[\text{cost}(s_1, \min_{i \neq 1} W_{\text{reqd}}(s_i))]$

$\mathbf{E}$[optimal **required**-inspection cost] $= \mathbf{E}[\min_i W_{\text{reqd}}(s_i)]$

newish $\mathbf{E}$[optimal **optional**-inspection cost] $\geq \mathbf{E}[\min_i W_{\text{optn}}(s_i)]$

$\mathbf{E}$[cost under **Local Hedging**] $= \mathbf{E}[\min_i W_{\text{LH}}(s_i)]$

$$W_{\text{LH}} = \begin{cases} W_{\text{reqd}} & \text{w.p. } p \\ W_{\text{grab}} & \text{w.p. } 1-p \end{cases}$$

NEW! **Lemma:** exists prob p s.t. for all r,

$$\mathbf{E}[\min\{\tfrac{3}{4} W_{\text{LH}}, r\}] \leq \mathbf{E}[\min\{W_{\text{optn}}, r\}]$$

"local"

Expressing cost using surrogate prices

s_1 s_2 s_3 …

$= \mathbf{E}[\text{cost}(s_1, \min_{i\neq 1} W_{\text{reqd}}(s_i))]$

$\mathbf{E}$[optimal **required**-inspection cost] $= \mathbf{E}[\min_i W_{\text{reqd}}(s_i)]$

newish $\mathbf{E}$[optimal **optional**-inspection cost] $\geq \mathbf{E}[\min_i W_{\text{optn}}(s_i)]$

$\mathbf{E}$[cost under **Local Hedging**] $= \mathbf{E}[\min_i W_{\text{LH}}(s_i)]$

$$W_{\text{LH}} = \begin{cases} W_{\text{reqd}} & \text{w.p. } p \\ W_{\text{grab}} & \text{w.p. } 1-p \end{cases}$$

NEW!

Lemma: exists prob p s.t. for all r,

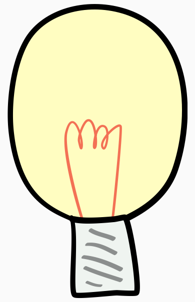
$$\mathbf{E}[\min\{\tfrac{3}{4}W_{\text{LH}}, r\}] \leq \mathbf{E}[\min\{W_{\text{optn}}, r\}]$$

"local"

"context"

Our contribution

Key idea: randomization for "context-robustness"



Local Hedging (LH)

New *decomposition-based* technique for optional inspection

- Reduces problem to required-inspection case
- Naturally generalizes to combinatorial problems

Theorem: if Alg is a "greedy" algorithm, then the approximation ratio of Gittins+Alg+LH is ≤ 4/3 times that of Alg

price of reduction from 3/4 discount